

**UNIVERSIDAD DEL NORTE**

**Departamento de Ingeniería Mecánica**



**DISEÑO DE UN MÉTODO ADAPTATIVO DE DETECCIÓN  
TEMPRANA DE FALLAS PARA LA REDUCCIÓN DE FALSAS  
ALARMAS DEBIDAS AL ENVEJECIMIENTO DE PROCESOS Y A  
LA INCERTIDUMBRE ESTADÍSTICA**

**TESIS**

Para la obtención del título de:

**Magíster en Ingeniería Mecánica**

**IVÁN DARÍO PORTNOY DE LA OSSA**

**Barranquilla, Colombia**

**Mayo 2014**

UNIVERSIDAD DEL NORTE

**Departamento de Ingeniería Mecánica**



DISEÑO DE UN MÉTODO ADAPTATIVO DE DETECCIÓN  
TEMPRANA DE FALLAS PARA LA REDUCCIÓN DE FALSAS  
ALARMAS DEBIDAS AL ENVEJECIMIENTO DE PROCESOS Y A  
LA INCERTIDUMBRE ESTADÍSTICA

TESIS

Para la obtención del título de:

Magíster en Ingeniería Mecánica

IVÁN DARÍO PORTNOY DE LA OSSA

Barranquilla, Colombia

Mayo 2014

## Tabla de Contenido

1. INTRODUCCIÓN .....	9
1.1. Planteamiento del Problema.....	9
1.2. Relevancia del Tema de Investigación.....	10
1.3. Objetivos de la Investigación.....	12
1.3.1. Objetivo General.....	12
1.3.2. Objetivos Específicos.....	12
2. REVISIÓN DEL ESTADO DEL ARTE .....	13
2.1. Técnicas de Detección de Fallas.....	13
2.1.1. Enfoque Analítico o Basado en Modelos .....	16
2.1.2. Enfoque Basado en Datos .....	17
2.1.3. Enfoque Basado en Conocimiento.....	22
2.2. Detección de Fallas Adaptativa.....	22
2.3. Inteligencia Artificial en Detección de Fallas .....	24
2.3.1. Técnicas de Detección de Fallas Basadas en Redes Neuronales Artificiales (ANNs) .....	25
2.3.2. Técnicas de Detección de Fallas Basadas en Lógica Difusa.....	26
2.3.3. Técnicas de Detección de fallas Basadas en Sistemas Expertos .....	27
2.3.4. Detección de Fallas Basada en Técnicas Híbridas de Inteligencia Artificial ....	28
3. DISEÑO DE UNA ESTRATEGIA ADAPTATIVA POR PCA RECURSIVO PONDERADO (WARP).....	29
3.1. PCA Convencional.....	30
3.1.1. Entrenamiento .....	30
3.1.2. Detección de Fallas .....	33
3.2. Desarrollo de Fórmulas Recursivas para PCA (RPCA).....	34
3.3. Desarrollo de Fórmulas Recursivas Ponderadas para PCA (WARP) .....	40
3.4. Actualización de Umbrales de Detección .....	43
3.5. Complejidad Computacional.....	44

4. DISEÑO DE LA ESTRATEGIA ADAPTATIVA CON UMBRAL DINÁMICO: MOVING AVERAGE FILTER PCA (MAF-PCA) .....	49
4.1. Estrategia Adaptativa con Umbral Dinámico para Estadístico $T^2$ .....	49
4.1.1. Experimento Factorial en la Región de Experimentación 1 (Para Estadístico $T^2$ )	51
4.1.2. Experimento Factorial en la Región de Experimentación 2 (Para Estadístico $T^2$ )	53
4.1.3. Optimización (Para Estadístico $T^2$ ) .....	55
4.2. Estrategia Adaptativa con Umbral Dinámico para Estadístico $Q$ .....	60
4.2.1. Experimento Factorial en la Región de Experimentación 1 (Para Estadístico $Q$ )	61
4.2.2. Experimento Factorial en la Región de Experimentación 2 (Para Estadístico $Q$ )	63
4.2.3. Optimización (Para Estadístico $Q$ ) .....	65
5. VALIDACIÓN DE TÉCNICAS DESARROLLADAS EN UN PROCESO DE REFERENCIA .....	73
5.1. Proceso de Referencia: Tennessee Eastman Process (TEP) .....	73
5.1.1. Descripción del Proceso .....	73
5.2. Validación de Estrategia WRPCA en Proceso de Referencia.....	75
5.3. Validación de Estrategia Adaptativa con Umbral Dinámico; Moving Average Filter PCA (MAF-PCA), en Proceso de Referencia .....	84
6. CONCLUSIONES .....	88
REFERENCIAS.....	90
ANEXOS .....	94
A.1. Esquema del Proceso .....	94
.....	94
A.2. Modelo del Proceso .....	94
A.3. Datos de Proceso .....	99
A.4. Nomenclatura .....	101
A.4. Código del Algoritmo WARP en MatLab® .....	104
A.5. Código del algoritmo MAF-PCA en MatLab® .....	107

## Listado de Figuras

Figura 1 - Etapas asociadas con el monitoreo de procesos (Adaptado de [3]).....	15
Figura 2 – Clasificación global de las técnicas de detección de fallas.....	16
Figura 3 - Carta de control de Shewhart. Tomado de [35]. .....	18
Figura 4 - Carta de control EWMA. Tomado de [35].....	20
Figura 5 - Gráfica de monitoreo (Run Chart) para PCA.....	21
Figura 6 – Clasificación de técnicas de detección de fallas basada en inteligencia artificial (Adaptado de [45]).....	24
Figura 7 – Evaluación de complejidad computacional, caso 1: $n_{k+1} = 20000$ , $n_{k+1}^* = 1000$ . .....	46
Figura 8 - Evaluación de complejidad computacional, caso 2: $n_{k+1} = 100000$ , $n_{k+1}^* = 5000$ . .....	46
Figura 9 - Evaluación de complejidad computacional, caso 3: $n_{k+1} = 1000000$ , $n_{k+1}^* = 50000$ . .....	47
Figura 10 – Evaluación de complejidad computacional incluyendo algoritmo de [5], caso 3: $n_{k+1} = 1000000$ , $n_{k+1}^* = 50000$ , escala logarítmica. ....	48
Figura 11 – Esquema del algoritmo MAF-PCA para el estadístico $T^2$ , con $r = 3$ . ....	50
Figura 12 - Gráfico de residuales para regresión de optimización para estadístico $T^2$ .....	58
Figura 13 – Respuesta del estadístico $T^2$ y del umbral en el punto óptimo de operación con MAF-PCA.....	59
Figura 14 - Respuesta del estadístico $T^2$ con PCA convencional. ....	59
Figura 15 – Esquema del algoritmo MAF-PCA para el estadístico $Q$ , con $r = 3$ .....	61
Figura 16 - Gráfico de residuales para regresión en la región de experimentación 1 para estadístico $Q$ . ....	67
Figura 17 - Gráfico de residuales de $k$ para regresión en la región de experimentación 2 para estadístico $Q$ . ....	69
Figura 18 - Gráfico de residuales de $r$ para regresión en la región de experimentación 2 para estadístico $Q$ . ....	70
Figura 19 – Respuesta del estadístico $Q$ y el umbral con el enfoque MAF-PCA. ....	71
Figura 20 – Respuesta del estadístico $Q$ con PCA convencional.....	72
Figura 21 – Diagrama de flujo del Tennessee Eastman Process (TEP). ....	74
Figura 22 – Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo WARP).....	77

Figura 23 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0$ . ....	78
Figura 24 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.1$ . ....	78
Figura 25 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.2$ . ....	79
Figura 26 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.3$ . ....	79
Figura 27 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.4$ . ....	80
Figura 28 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.5$ . ....	80
Figura 29 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.6$ . ....	81
Figura 30 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.7$ . ....	81
Figura 31 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.9$ . ....	82
Figura 32 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.95$ . ....	82
Figura 33 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=0.99$ . ....	83
Figura 34 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con $w=1$ . ....	83
Figura 35 – Desempeño del algoritmo WARP en función del factor de aprendizaje $w$ . ....	84
Figura 36 - Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo MAF-PCA). ....	85
Figura 37 - Desempeño del algoritmo MAF-PCA ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200. ....	85
Figura 38 - Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo MAF-PCA). ....	86
Figura 39 - Desempeño del algoritmo MAF-PCA ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200. ....	86

## Listado de Tablas

Tabla 1 - Comparación de la complejidad computacional de actualización recursiva vs actualización no recursiva.....	45
Tabla 2 - Factores y niveles definidos para la región de experimentación 1 (Para Estadístico $T^2$ ).....	52
Tabla 3 - Condiciones experimentales de la región de experimentación 1 (Para Estadístico $T^2$ ).....	52
Tabla 4 - ANOVA para la región de experimentación 1 (Para Estadístico $T^2$ ).....	53
Tabla 5 - Factores y niveles definidos para la región de experimentación 2 (Para Estadístico $T^2$ ).....	54
Tabla 6 - Condiciones experimentales de la región de experimentación 2 (Para Estadístico $T^2$ ).....	54
Tabla 7 - ANOVA para la región de experimentación 2 (Para Estadístico $T^2$ ).....	55
Tabla 8 - Resumen de modelos de regresión para optimización de la tasa de falsas alarmas para el estadístico $T^2$ .....	56
Tabla 9 – Estadísticas de la regresión para optimización para estadístico $T^2$ .....	56
Tabla 10 - ANOVA de la regresión de la tasa de falsas alarmas para estadístico $T^2$ .....	57
Tabla 11 - Análisis de los coeficientes de regresión para optimización para estadístico $T^2$ .....	57
Tabla 12 - Factores y niveles definidos para la región de experimentación 1. (Para estadístico $Q$ ).....	62
Tabla 13 - Condiciones experimentales de la región de experimentación 1 (Para Estadístico $Q$ ).....	62
Tabla 14 - ANOVA para la región de experimentación 1 (Para Estadístico $Q$ ).....	63
Tabla 15 - Factores y niveles definidos para la región de experimentación 2. (Para estadístico $Q$ ).....	63
Tabla 16 - Condiciones experimentales de la región de experimentación 2 (Para Estadístico $Q$ ).....	64
Tabla 17 - ANOVA para la región de experimentación 2 (Para Estadístico $Q$ ).....	65
Tabla 18 - Resumen de modelos de regresión para optimización de la tasa de falsas alarmas para estadístico $Q$ .....	66
Tabla 19 - Estadísticas de la regresión en la Región de experimentación 1 para estadístico $Q$ .....	66
Tabla 20 - ANOVA de la regresión de la tasa de falsas alarmas para el estadístico $Q$ en la región de operación 1.....	66

Tabla 21 - Análisis de los coeficientes de regresión en la Región de experimentación 1 para estadístico $T^2$ .....	67
Tabla 22 - Modelos de regresión para la tasa de falsas alarmas en la Región de experimentación 2 para estadístico Q. ....	68
Tabla 23 - Estadísticas de la regresión en la Región de experimentación 2 para estadístico Q. ....	68
Tabla 24 - ANOVA de la regresión de la tasa de falsas alarmas para el estadístico Q en la región de operación 1. ....	69
Tabla 25 - Análisis de los coeficientes de regresión en la Región de experimentación 2 para estadístico $T^2$ .....	69
Tabla 26 – Pares de variables cuyo coeficiente de correlación .....	76
Tabla 27 – Variables excluidas. ....	76



# **1. INTRODUCCIÓN**

En este capítulo se evidencian las limitaciones de las técnicas de detección de fallas convencionales cuando son implementadas para el monitoreo de procesos altamente complejos y no-lineales que tengan variables altamente correlacionadas y que muestren cambios de varianzas de las mismas a lo largo del tiempo (cambiando la estructura estadística subyacente a los mismos). De esta manera, se plantea el problema de investigación, resaltándose posteriormente la relevancia del mismo. Se plantean finalmente objetivos, general y específicos, cuyo cumplimiento esté enfocado a la resolución del problema de investigación planteado.

## **1.1. Planteamiento del Problema**

Desde los años 60 se ha incrementado de manera progresiva la instrumentación de procesos, lo que ha incrementado el nivel de monitoreo de variables y ha permitido una mayor disponibilidad de datos históricos y datos en línea del proceso [2]. Debido a lo anterior se ha extendido el uso de las técnicas de monitoreo estadístico de procesos en lugar de técnicas basadas en modelos, las cuales requieren modelos matemáticos con una descripción muy precisa de los procesos. Dichos modelos con la precisión requerida para la detección de fallas resultan costosos de desarrollar para procesos altamente complejos y no-lineales. Las cartas de control y las cartas EWMA y CUSUM son ejemplos de técnicas univariadas de monitoreo estadístico de procesos. Existen también técnicas multivariadas de monitoreo de procesos (MSPC) tales como el análisis de componentes principales (PCA), una de las más ampliamente utilizadas [3, 10, 23, 24, 40].

Una desventaja notable de los métodos de MSPC, incluyendo a PCA, es que implícitamente asumen que las observaciones en un determinado instante son estadísticamente independientes de otras observaciones en instantes anteriores o futuros, por lo que una vez estructurados los modelos de detección de fallas en la fase de entrenamiento (con los datos históricos de proceso) éstos permanecen invariantes, mientras que la mayoría de los procesos reales muestran cambios de comportamiento en el tiempo [25, 29, 30, 40]. Cuando PCA es aplicado con modelos de detección de fallas fijos para el monitoreo de procesos con cambios o derivas temporales normales, las falsas alarmas eventualmente ocurrirán.

Como se ha establecido, los procesos reales presentan cambios de varianzas de las variables a lo largo del tiempo, cambiando consecuentemente la estructura estadística subyacente a los mismos. Se presentan además cambios aleatorios sutiles que pueden desencadenar falsas alarmas eventuales. Debido a esto, se requiere el diseño de una estrategia basada en PCA con el fin de reducir la tasa de falsas alarmas causadas por el envejecimiento del proceso e incertidumbre estadística, reduciendo además la complejidad de la implementación de la adaptación en línea respecto a alternativas similares existentes en el estado del arte [5, 20].

## **1.2. Relevancia del Tema de Investigación**

Las plantas de las industrias modernas, en especial las de la industria química y de hidrocarburos, son de gran escala, altamente complejas y operan con un gran número de variables en lazos cerrados de control [3]. Todos los procesos son susceptibles a experimentar condiciones de operación inadecuadas o fallas y si no se lleva a cabo un monitoreo continuo de estos procesos dichas fallas sólo son detectables una vez que tengan consecuencias significativas. Consecuentemente, se deben llevar a cabo esfuerzos para reducir las pérdidas de productos debido a la detección tardía o no detección de fallas, así como también para satisfacer las regulaciones de seguridad y medio ambientales. El enfoque con el que se ha abordado tradicionalmente el monitoreo de procesos son las cartas de control, las cuales siguen el comportamiento variable a variable y establecen límites prismáticos de comportamiento de cada una de éstas, considerándose como comportamientos atípicos a cualquier violación de dichos límites. Sin embargo estas técnicas ignoran correlaciones entre las distintas variables y auto-correlaciones de las mismas, por lo que son incapaces de monitorear adecuadamente procesos con variables fuertemente correlacionadas y con no-linealidades [1, 3, 5].

Dada la complejidad subyacente a muchos de los procesos o plantas presentes en la industria, un enfoque de detección de fallas basado en modelos resulta poco práctico si se va a implementar en éstos ya que se necesitarían modelos que los describan de manera muy precisa [5]. Por otro lado, desde los años 60 se ha incrementado de manera progresiva la instrumentación de procesos, lo que ha promovido el monitoreo de los procesos y permite la disponibilidad tanto de datos históricos como datos en línea del proceso [2]. Consecuentemente, un enfoque de detección de fallas basado en

datos resulta apropiado si se tiene como objetivo monitorear un proceso de alta complejidad. Con la disponibilidad de datos de proceso históricos y en línea se podría también usar un enfoque de detección de fallas basado en conocimiento, sin embargo la complejidad computacional de las técnicas basadas en datos es menor, lo que las hace una opción más práctica para este tipo de procesos [3].

El análisis de componentes principales (PCA) es una técnica, basada en datos, de reducción dimensional usada para llevar a cabo detección de fallas la cual captura de manera óptima la varianza de los datos en un espacio de menor dimensión y la cual ha sido ampliamente utilizada para el monitoreo de procesos [10, 23, 24]. La gran cantidad de variables medidas pueden ser representadas, mediante PCA, con un simple parámetro estadístico, tal como lo es el estadístico  $T^2$  [26, 27] o el estadístico  $Q$  [12], los cuales pueden ser mostrados en una sola gráfica con un umbral estadístico, de manera análoga a lo que se hace con las cartas de control con cada una de las variables [28]. PCA requiere un entrenamiento inicial, etapa en la cual se fija un modelo estadístico de detección así como también se fijan los umbrales para los estadísticos  $T^2$  y  $Q$  respectivamente. Ya que esta técnica implícitamente asume que las observaciones en un determinado instante son estadísticamente independientes de otras observaciones en instantes anteriores o futuros, no considera eventuales cambios en la estructura de varianzas del proceso en el tiempo. Sin embargo, los procesos industriales reales muestran sutiles cambios de comportamiento en el tiempo, tales como: desactivación catalítica, ensuciamiento de intercambiadores de calor, taponamiento gradual de válvulas y envejecimiento de equipos, sensores y del proceso mismo [25, 29, 30].

La literatura revisada concerniente al monitoreo de procesos y detección de fallas sugiere que se requieren métodos que tengan en cuenta las variaciones del comportamiento de procesos en el tiempo y auto-correlaciones en las variables con el fin de alcanzar un esquema más completo de monitoreo de procesos [5, 31]. Algunos de los enfoques que buscan tener en cuenta las auto-correlaciones en las variables son las cartas de monitoreo con media móvil, las cartas EWMA, las cartas CUSUM integradas con PCA y el PCA dinámico o DPCA [31, 32, 33, 34]. Existen además técnicas basadas en PCA que tienen en cuenta nuevos datos de proceso e incorporan su información en los modelos estadísticos de detección [3, 5, 20]. La inclusión de nueva información disponible de operación de procesos y la consideración de auto-correlaciones en las variables de los mismos resultan ser pertinentes si se quiere construir un esquema de monitoreo de procesos más completo. Consecuentemente, el

desarrollo de nuevas estrategias que utilicen e integren estos dos aspectos en consideración, así como también la mejora de las técnicas existentes constituyen una línea de investigación relevante.

### **1.3. Objetivos de la Investigación**

En esta sección se define el objetivo general, así como también los objetivos específicos de la investigación.

#### **1.3.1. Objetivo General**

Diseñar técnicas adaptativas de detección temprana de fallas basadas en análisis de componentes principales (PCA) con el fin de reducir la tasa de falsas alarmas causadas por el envejecimiento del proceso e incertidumbre estadística, reduciendo además la complejidad de la implementación de la adaptación en línea.

#### **1.3.2. Objetivos Específicos**

- Desarrollar una técnica adaptativa de detección de fallas que reduzca la tasa de falsas alarmas mediante la reestructuración de modelos estadísticos con la inclusión de nuevos datos disponibles de operación normal de proceso y evidencia de falsas alarmas, generando una solución analítica equivalente.
- Desarrollar una técnica adaptativa de detección de fallas que reduzca la tasa de falsas alarmas mediante la implementación de una ventana móvil a los estadísticos de monitoreo y la definición de reglas de decisión para fijar los umbrales de detección.

## 2. REVISIÓN DEL ESTADO DEL ARTE

Se lleva a cabo en este capítulo una revisión del estado del arte, la cual se divide en tres secciones. En la primera sección se abordarán los distintos enfoques encontrados en la literatura para la detección de fallas, lo que conduce a realizar una clasificación de las técnicas existentes. En la segunda sección se revisan las técnicas adaptativas de detección de fallas, i.e., aquellas técnicas que tienen en cuenta las variaciones de los procesos en el tiempo. Se hace finalmente, en la sección tres, una revisión de las técnicas de inteligencia artificial aplicadas a la detección y diagnóstico de fallas en procesos.

### 2.1. Técnicas de Detección de Fallas

Tradicionalmente la etapa de detección de fallas en el monitoreo de procesos ha consistido en hacer seguimiento de cada una de las variables de proceso cuyo comportamiento afecte de manera relevante al mismo. Dicho seguimiento consiste en verificar que cada variable observada, o algún parámetro asociado a la misma, se mantengan dentro de unos rangos permitidos en los que su comportamiento se considera normal [3].

Los métodos estadísticos univariados tradicionales, como lo son las cartas de Shewhart, consisten en el establecimiento de umbrales basados en un estadístico de prueba que no deben ser sobrepasados por un estadístico calculado para cada observación de la variable monitoreada. Se establece entonces una regla de rachas de valores fuera de los umbrales para la detección de fallas. Esta metodología es denominada en inglés *Limit Sensing* y ha sido ampliamente aplicada debido a su facilidad de implementación y comprensión, sin embargo carece de sensibilidad en procesos complejos ya que no considera correlaciones espaciales entre las distintas variables del proceso así como tampoco considera correlaciones temporales [2, 3]. Estas correlaciones son ignoradas porque este enfoque determina umbrales para cada variable del proceso sin usar ninguna información de las otras variables y además no actualiza estos umbrales. Otro enfoque tradicional, llamado *Detección por Discrepancia*, consiste en comparar el valor actual medido de cada variable con un valor correspondiente de un modelo analítico y si son significativamente diferentes se activa una alarma indicando que ha sido detectada una falla. Sin embargo, este

enfoque requiere que se tenga un modelo analítico que describe de manera precisa el proceso, lo cual resulta difícil de conseguir cuando se trata de procesos complejos y altamente no lineales [6]. Por lo tanto, para el monitoreo de procesos altamente complejos que presenten altas correlaciones entre sus variables y no-linealidades, resulta necesaria la implementación de técnicas que tengan en cuenta dichas correlaciones y no linealidades, tales como las técnicas basadas en datos para la detección de fallas, e.g. PCA (Principal Component Analysis).

Las cuatro etapas asociadas con el monitoreo de procesos son:

- La ***Detección de Fallas***, que es una acción binaria que consiste en determinar una falla ha o no ocurrido.
- La ***Identificación de Fallas***, que consiste en determinar cuáles variables son más significativas para el diagnóstico de la falla.
- El ***Diagnóstico de Fallas***, consiste en determinar el tipo, ubicación, magnitud y tiempo de ocurrencia de la falla.
- La ***Recuperación del Proceso***, que consiste en tomar acciones correctivas para llevar de nuevo al proceso a una condición de operación normal.

En la *Figura 1* se muestra un esquema de las cuatro etapas asociadas con el monitoreo de procesos en un diagrama de flujo.

Un esquema típico de monitoreo de procesos contiene una o más medidas, que pueden estar basadas en los desarrollos de teorías estadísticas, de modelos, de teorías de reconocimiento de patrones, etc. Estas medidas son calculadas directamente de los datos de proceso y requieren por lo tanto que exista una instrumentación adecuada. Se busca convertir la gran cantidad de datos en unas cuantas medidas significativas del status del proceso y así asistir a los operadores a detectar e identificar fallas en los procesos.

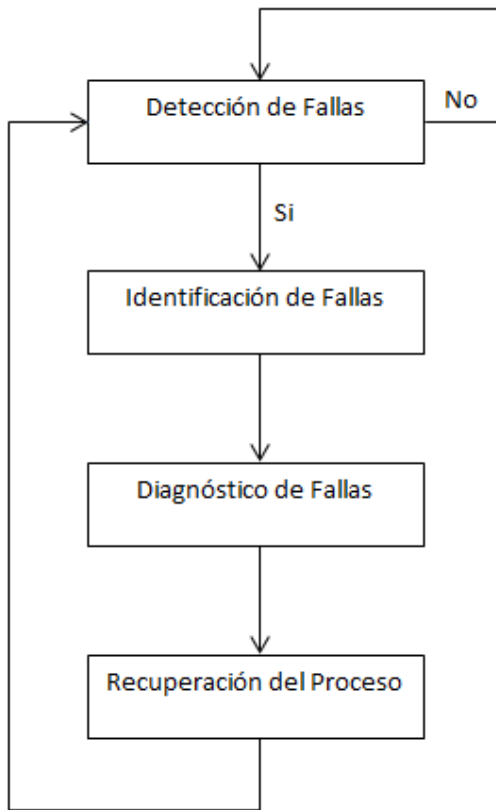


Figura 1 - Etapas asociadas con el monitoreo de procesos (Adaptado de [3]).

Se derivan entonces distintos enfoques con los que se puede abordar la detección y/o diagnóstico de fallas. Se han identificado principalmente de tres enfoques; llamados, ***basado en datos, analítico (o basado en modelos)*** y ***basado en conocimiento***. Las medidas basadas en datos son derivadas directamente de los datos de proceso, contrastando con el enfoque analítico que usa modelos matemáticos usualmente contruidos basándose en principios físicos, mientras que el enfoque basado en conocimiento emplea modelos cualitativos. El enfoque analítico y el basado en conocimiento tienen su mayor aplicabilidad en sistemas con un número relativamente pequeño de variables y estados [3].

Se muestra en la *Figura 2* un esquema de la clasificación global de las técnicas de detección de fallas, con algunos ejemplos de técnicas pertenecientes a cada uno de los tres enfoques previamente mencionados.

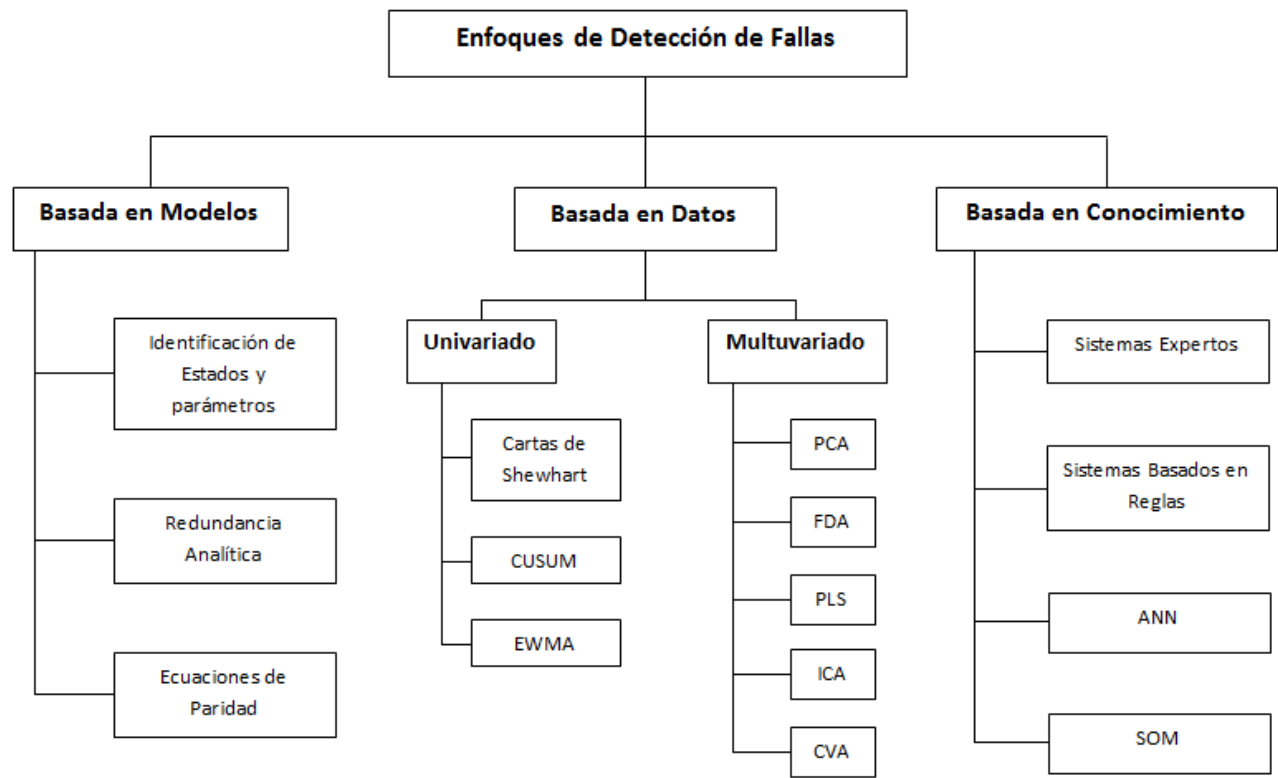


Figura 2 – Clasificación global de las técnicas de detección de fallas.

Se presenta a continuación una vista general de estos enfoques.

### 2.1.1. Enfoque Analítico o Basado en Modelos

Este enfoque fue hasta 1997 el más ampliamente usado para llevar a cabo la detección y diagnóstico de fallas [1]. Este puede brindar un mejor monitoreo de procesos cuando se cuenta con modelos basados en principios fundamentales que describan de manera precisa dichos procesos, siendo en tal caso adecuado para la detección y diagnóstico de fallas. No obstante, mientras más grande es el número de variables de procesos y más complejidad tenga el mismo, más difícil es su modelamiento matemático. Este enfoque puede estar basado en tres sub enfoques; *Estimación de Parámetros*, *Estimación de Estados* o *Redundancia Analítica*.



#### **2.1.1.1.    *Estimación de estados***

Es apropiado si las fallas del proceso están asociadas con cambios en variables de estado no medibles. Los estados son reconstruidos de variables medibles de entrada y salida del proceso usando un observador de estado [3]. Se puede establecer umbrales para algunos o todos los cambios en los estados estimados.

#### **2.1.1.2.    *Estimación de parámetros***

Es apropiado si las fallas del proceso están asociadas con cambios de parámetros en el modelo del proceso. La construcción de modelos basados en principios fundamentales permite relacionar directamente los parámetros del modelo con parámetros que tienen significado físico. Se puede establecer umbrales para la diferencia individual entre los parámetros nominales de proceso y los parámetros estimados. [3, 52, 53]

.

#### **2.1.1.3.    *Redundancia Analítica***

Incorpora un modelo explícito del proceso para generar y evaluar residuales [3, 52]. En el caso de estimación de parámetros, estos residuales pueden ser tomados como la diferencia entre los parámetros nominales de proceso y los parámetros estimados, mientras que en el caso de estimación de estados los residuales pueden ser tomados como la diferencia entre las variables nominales de estado y las variables de estado estimadas. Los residuales deben ser evaluados siendo usados como entradas para la detección y diagnóstico de fallas.

### **2.1.2. Enfoque Basado en Datos**

Las técnicas de detección de fallas basadas en datos se dividirán en dos categorías principales: Técnicas univariadas y técnicas multivariadas, las cuales se presentan brevemente en esta sección

### 2.1.2.1. Técnicas Univariadas

Las principales técnicas univariadas basadas en datos encontradas en la literatura se describen a continuación.

#### *Cartas de Shewhart*

Las cartas de Shewhart constituyen una técnica de control estadístico de procesos (SPC) que se basa en la suposición de que un proceso sujeto a una variabilidad natural se mantendrá en un estado de control estadístico bajo el cual ciertas variables de proceso se mantienen cerca de sus valores deseados. Manteniendo la suposición anterior se puede, mediante la supervisión del comportamiento del proceso, detectar los eventos anormales tan pronto como se produzcan. Si las causas de ese tipo de eventos (fallas) se pueden diagnosticar y el problema se puede corregir, el proceso puede ser conducido de nuevo a su funcionamiento normal [49]. Se observa en la *Figura 3* un ejemplo de carta de Shewhart.

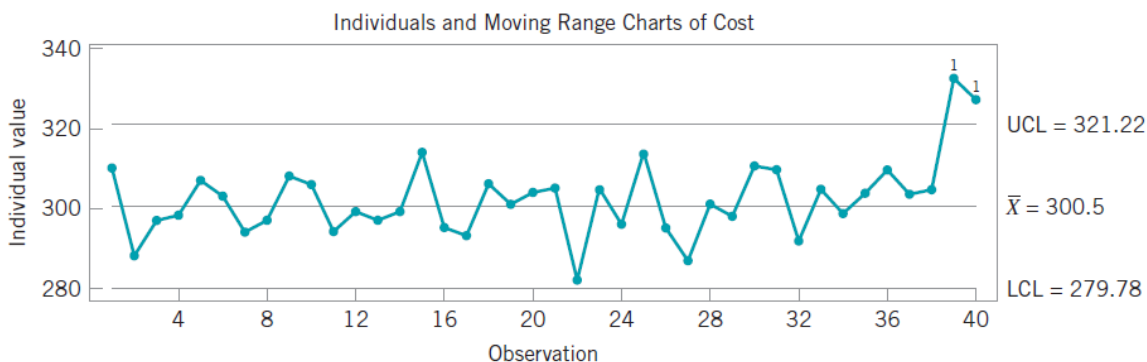


Figura 3 - Carta de control de Shewhart. Tomado de [35].

En la *Figura 3* los términos UCL y LCL hacen referencia al límite superior de control y al límite inferior de control respectivamente, mientras que  $\bar{X}$  hace referencia a la media de la variable que se está monitoreando. Cualquier observación que se encuentre por fuera de la región acotada por el UCL y el LCL es una *condición atípica de operación* o condición “fuera de control”, tal como las observaciones de la 35 a la 40 de la *Figura 3*. Al tipo de gráfica que se usa para presentar las cartas de

Shewhart (así como también para CUSUM y EWMA) se le denomina en inglés ***Run Chart***.

### ***CUSUM***

La carta de control CUSUM, o “*Carta de control de Suma Acumulativa*” por sus siglas en inglés, es una técnica de análisis secuencial que busca monitorear la variabilidad de un parámetro de calidad  $\theta$ , el cual puede ser por ejemplo la varianza o la media de una determinada variable de proceso, con el fin de detectar eventuales cambios en la misma. Este método requiere que se conozca, o se estime, el parámetro de calidad de la variable a monitorear. Como su nombre lo indica, implica el cálculo de una suma acumulativa de muestras de la variable de proceso a las cuales se les asignan pesos [34, 35, 39].

Al igual que las cartas de Shewhart, la gráfica de monitoreo utilizada por CUSUM comprende un límite inferior LCL y un límite superior UCL y se lleva a cabo el monitoreo mediante una ***Run Chart***.

### ***EWMA***

EWMA o media móvil ponderada exponencialmente (Exponentially Weighted Moving Average) es un tipo de carta de control utilizada para monitorear variables de proceso (una a una) utilizando como insumo los datos históricos de proceso disponibles [33, 38]. EWMA pondera las muestras en un orden geométricamente decreciente de manera que a las muestras más recientes se les asigna un peso grande mientras que a aquellas muestras más distantes contribuyen menos al cálculo de la media móvil.

Al igual que las cartas de Shewhart, la gráfica de monitoreo utilizada por EWMA comprende un límite inferior LCL y un límite superior UCL, los cuales son móviles, difiriendo así de los usados en las cartas de Shewhart. Se lleva a cabo también el monitoreo mediante una ***Run Chart***, como se ilustra en la *Figura 4*.

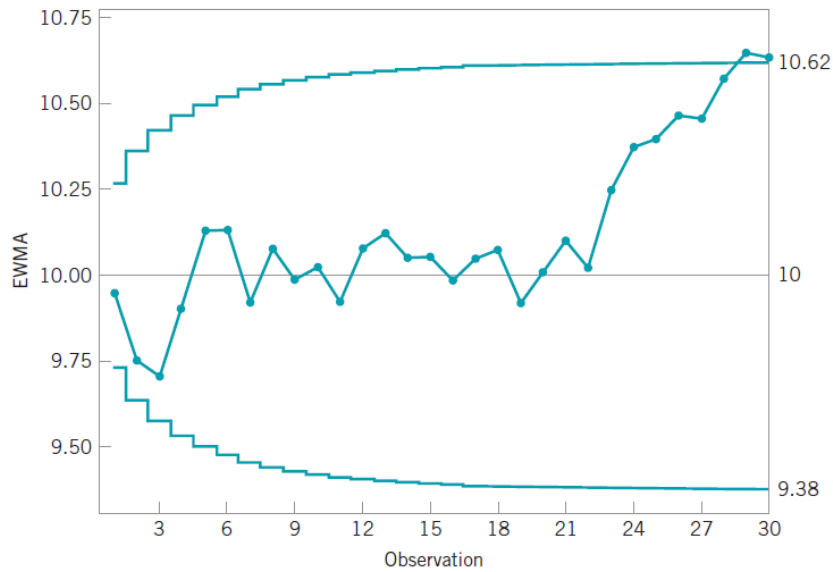


Figura 4 - Carta de control EWMA. Tomado de [35].

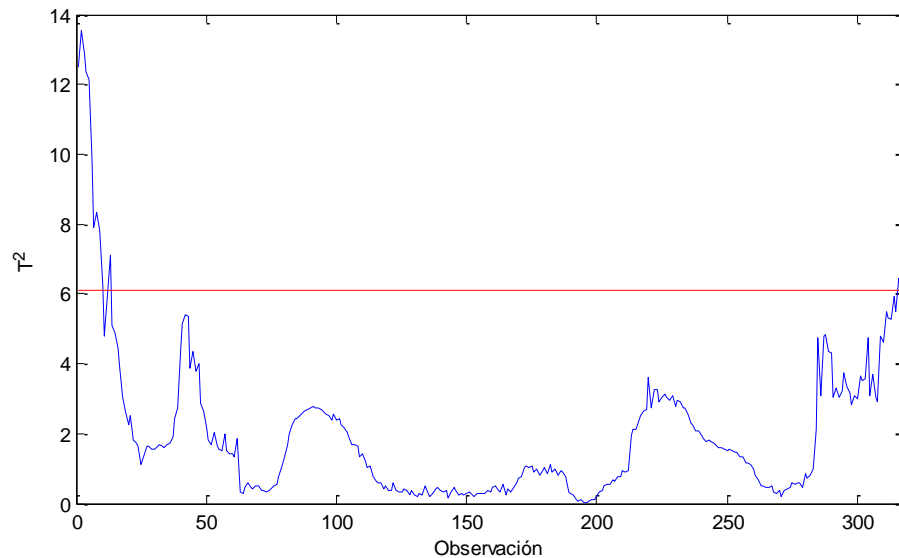
#### 2.1.2.2. Técnicas Multivariadas

Ya que las técnicas tradicionales basadas en datos, como las cartas de Shewhart, CUSUM, EWMA, etc. pierden robustez al ignorar las correlaciones temporales y espaciales [2], algunas técnicas de monitoreo de procesos basadas en estadística multivariada han sido desarrolladas para superar esta dificultad. Se resumen a continuación las más relevantes de estas técnicas.

##### *Principal Component Analysis (PCA)*

PCA es una técnica de reducción dimensional que captura óptimamente la variabilidad de los datos en un espacio de pequeña dimensión que es ampliamente usada en el monitoreo de procesos [8, 9, 10]. PCA permite además el conocimiento de las variables principalmente responsables de las fallas así como de las más afectadas por la misma [5]. El gran número de variables es convertido en mediciones simples tal como los estadísticos  $T^2$  de Hotelling y  $Q$ , los cuales puede ser mostrados cada uno en una gráfica simple y comparados con un umbral estadístico tal y como se hace en las cartas univariadas [12]. Existen algunos enfoques que mejoran esta

técnica como lo es PCA dinámico [9], algoritmos PCA recursivos [5], o combinándolo con lógica difusa y redes neuronales [10, 11]. Se muestra en la *Figura 5* un esquema de monitoreo (**Run Chart**) típico de PCA.



**Figura 5 - Gráfica de monitoreo (Run Chart) para PCA.**

Se observa que se ha graficado el estadístico  $T^2$  (en azul) y el umbral estadístico  $T_\alpha^2$  (en rojo), el cual es análogo a los límites de control en las cartas de control de Shewhart, CUSUM y EWMA, pero en este caso existe sólo un límite superior ya que el estadístico  $T^2$  cumple:  $T^2 \geq 0$ . Se lleva a cabo una gráfica similar para el estadístico  $Q$ . Las demás técnicas multivariadas que se presentan a continuación usan gráficas de monitoreo similares a la usada por PCA.

### ***Fisher Discriminant Analysis (FDA)***

FDA es también una técnica de reducción dimensional que es óptima para maximizar la separación entre distintas clases de comportamientos (por ejemplo fallas). FDA tiene en cuenta la información entre las distintas clases [14]. FDA brinda una representación de pequeña dimensión óptima en términos de discernir entre las distintas clases de fallas.

### ***Independent Component Analysis (ICA)***

Es también una técnica de reducción de dimensión y permite el uso de estadísticos para el monitoreo en línea y diagnóstico de fallas, y no requiere que los datos sigan alguna distribución en especial. ICA extrae los componentes ocultos de los datos asumiendo solamente que los factores desconocidos son mutuamente independientes [13].

### ***Partial Least Squares (PLS)***

Es, al igual que PCA, FDA e ICA, una técnica de reducción de dimensión, la cual busca maximizar la covarianza entre un bloque predictor  $X$  y bloque predicho  $Y$  para cada componente. Esta técnica modela la relación entre  $X$  y  $Y$  usando de ajustes locales por medio de mínimos cuadrados [3, 19].

#### **2.1.3. Enfoque Basado en Conocimiento**

Este enfoque utiliza la relación entre el patrón de comportamiento de los datos y las clases de fallas sin modelar o los estados internos del proceso o su estructura explícitamente [3]. Esto incluye los *sistemas expertos*, *sistemas basados en reglas*, *redes neuronales artificiales (ANN)*, *mapas auto-organizados (SOM)*, *árboles de decisión*, etc [4].

Se hablará en este mismo capítulo, en una sección posterior, de técnicas de inteligencia artificial aplicadas a la detección de fallas, por lo cual no se tratan en profundidad estas técnicas ahora.

## **2.2. Detección de Fallas Adaptativa**

Debido a que los algoritmos que se van a desarrollar estarán enfocados al monitoreo de procesos de alta complejidad, se expondrán los enfoques adaptativos de detección de fallas concernientes al monitoreo estadístico multivariado de procesos (MSPC). Una desventaja importante de los métodos de MSPC, dentro de los cuales PCA es

uno de los más usados [3, 40], es que una vez estructurados los modelos de detección de fallas en la fase de entrenamiento (con los datos históricos de proceso) éstos permanecen invariantes, mientras que la mayoría de los procesos reales muestran cambios de comportamiento en el tiempo [25, 29, 30, 40]. La inclusión de nueva información disponible de operación de procesos y la consideración de auto-correlaciones en las variables de los mismos resultan ser pertinentes si se quiere construir un esquema de monitoreo de procesos más completo.

Cuando PCA es aplicado con modelos de detección de fallas fijos para el monitoreo de procesos con cambios o derivas temporales normales, las falsas alarmas eventualmente ocurrirán. Consecuentemente se han desarrollado algoritmos de detección de fallas enfocados en lograr detección de fallas adaptativa para disminuir las falsas alarmas [40]. Un método sensible a la varianza basado en PCA es propuesto en la literatura y ha sido combinado con un umbral adaptativo para detección de fallas en un proceso experimental, el cual se caracteriza por mitigar las falsas alarmas ocasionadas por los comportamientos transitorios de los procesos mediante la captura y actualización constante de la varianza de del estadístico  $T^2$  para el ajuste del umbral de detección [39]. Algunos enfoques buscan tener en cuenta auto-correlaciones en las variables, tales como las cartas EWMA, las cartas CUSUM integradas con PCA y el PCA dinámico o DPCA [31, 32, 33, 34]. Existen además técnicas basadas en PCA que buscan la incorporación periódica de nuevos datos de proceso, actualizando esta información en los modelos estadísticos de detección. Tal es el caso de los enfoques que llevan a cabo actualizaciones recursivas de los modelos estadísticos de detección de fallas [5, 20, 40, 43], así como el enfoque de actualización recursiva combinada con ventana móvil hecha por X. Liu et al [41]. También se encuentran en la literatura algoritmos recursivos de detección de fallas basados en técnicas distintas a PCA, tal como los algoritmos propuestos por S.J. Qin [42], los cuales están basados en PLS (Partial Least Squares).

Además de los cambios en el tiempo que puede presentar un proceso, pueden tener distintos modos o patrones de operación, por lo que se debe, en estos casos, usar técnicas de reconocimiento de patrones en la fase de identificación y diagnóstico de fallas, para discernir entre distintos tipos de condiciones normales y condiciones de falla [16, 17, 18].

En el panorama nacional, se han encontrado en el estado del arte y la técnica escasos casos de desarrollos e implementaciones de técnicas de detección y diagnóstico de fallas [7, 15], los cuales ha sido desarrollados particularmente para redes de

transporte de hidrocarburos [15] y uno de los cuales [7] aborda el problema de aplicación en procesos con patrones de comportamiento.

### 2.3. Inteligencia Artificial en Detección de Fallas

Las técnicas de inteligencia artificial son aquellas que buscan el diseño de entidades capaces de procesar información usando el paradigma del razonamiento humano y las cuales tienen diversas aplicaciones, siendo la detección y el diagnóstico de fallas ejemplos de las mismas.

En la revisión de la literatura se encuentra que diversas técnicas de inteligencia artificial han sido aplicadas exitosamente en la detección y diagnóstico de fallas en procesos [45]. La *Figura 6* muestra una clasificación de las técnicas de detección de fallas que utilizan técnicas de inteligencia artificial (IA).

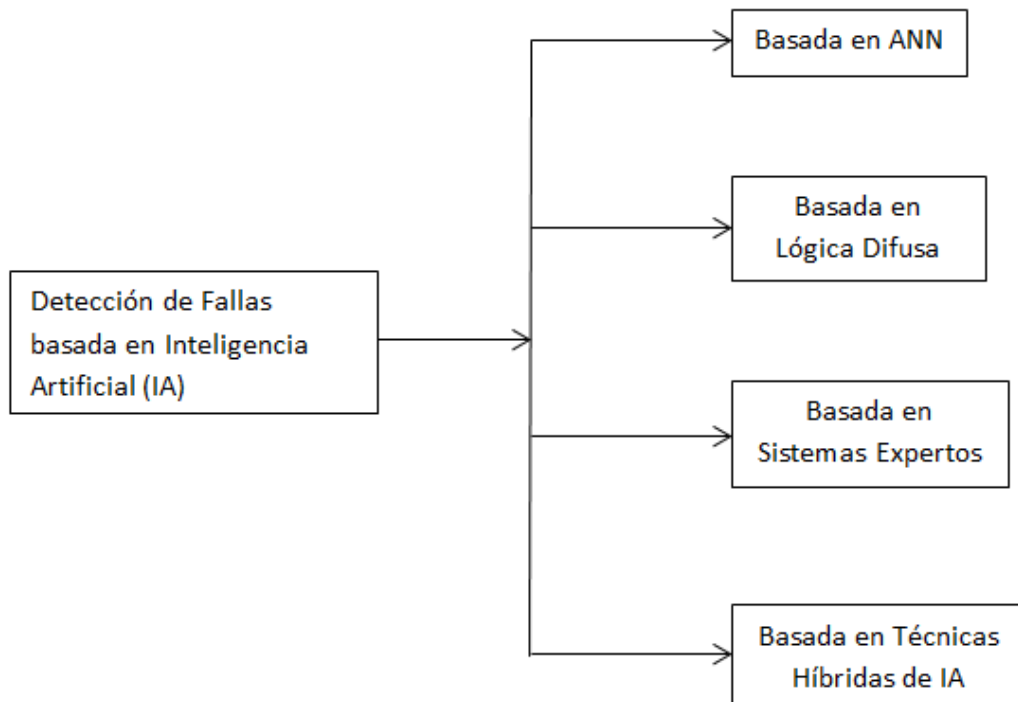


Figura 6 – Clasificación de técnicas de detección de fallas basada en inteligencia artificial (Adaptado de [45]).



Se describe a continuación de manera breve cada uno de los principales enfoques de detección de fallas basada en técnicas de inteligencia artificial, exponiéndose algunos ejemplos de sus aplicaciones.

### **2.3.1. Técnicas de Detección de Fallas Basadas en Redes Neuronales Artificiales (ANNs)**

Una red neuronal es un paradigma para el procesamiento de información que está inspirada en la forma en la que las redes neuronales biológicas, tales como el cerebro humano, funcionan. El bloque central de dicho paradigma es la estructura única de procesamiento de información, la cual consiste en un gran número de elementos de procesamiento altamente interconectados y organizados, los cuales se llaman *neuronas*. Una red neuronal artificial (ANN) usualmente es implementada para desempeñar una tarea en particular, tal como reconocimiento de patrones, clasificación y agrupamiento de datos, así como también detección y diagnóstico de fallas a través de un proceso de aprendizaje.

S.R. Samantaray [51] et al proponen una combinación de un filtro de Kalman con una red neuronal artificial para llevar a cabo detección de fallas en sistemas de distribución de energía. El desarrollo propuesto en [51] una red neuronal de perceptrón multicapa, también conocida como Feed-forward Neural Network (FNN), que consta de múltiples ( $n$ ) entradas y dos salidas, así como también una capa oculta con cinco neuronas ocultas.

Z.N. Sadough Vanini et al [48] implementan un esquema de detección de fallas para una turbina de gas de doble eje. El sistema que se propone en [48] utiliza redes neuronales dinámicas (DNNs), las cuales son una modificación de las redes neuronales artificiales convencionales (ANNs). Las DNNs son construidas para aprender la dinámica no lineal del proceso. Cada DNN implementada corresponde a un modo específico de operación. Las redes neuronales dinámicas o red dinámica de perceptrón multicapa (MLP) representa una extensión de las redes neuronales estáticas mediante la incorporación de un bloque dinámico en el modelo neuronal, con lo cual se pueden abordar no sólo las no linealidades estáticas sino también las dinámicas.

Se han presentado sólo un par de ejemplos de aplicaciones de redes neuronales artificiales (ANNs) para llevar a cabo la detección y diagnóstico de fallas, sin embargo existen en la literatura muchos otros enfoques que son también basados en ANNs.

### **2.3.2. Técnicas de Detección de Fallas Basadas en Lógica Difusa**

La lógica difusa provee un método para reducir y explicar la complejidad de sistemas, mitigar su incertidumbre y ambigüedades imitando el proceso de razonamiento humano. La lógica difusa permite que la pertenencia de variables de ingeniería a un grupo o conjunto sea estimada con un determinado grado de incertidumbre. En este sentido, la aplicación de lógica difusa para la detección y diagnóstico de fallas en procesos debería definir el grado de pertenencia de un nuevo espectro de señales (una nueva muestra) a los distintos espectros que han sido usados para definir los modos normales de operación así como también las fallas conocidas [45]. Se presentarán dos casos de enfoques de detección y/o diagnóstico de fallas en procesos.

L.J de Miguel et al [46] proponen una técnica de toma de decisiones basada en lógica difusa para la detección y diagnóstico de fallas que puede compensar la incertidumbre de los modelos de procesos, la presencia de ruido y el comportamiento estocástico de las variables de proceso. El módulo de toma de decisiones está basado en una serie de reglas difusas SI-ENTONCES (IF-THEN) y funciones de pertenencia, definidas para las ecuaciones de los residuales.

A. Lemos et al [47] proponen un enfoque adaptativo de detección y diagnóstico de fallas el cual detecta nuevos modos de operación así como también fallas e incorpora información sobre modos de operación en un clasificador de evolución difusa usado para el diagnóstico. El enfoque se basa en un procedimiento de agrupamiento (clustering) incremental para generar reglas difusas que describen nuevos estados operativos detectados.

### **2.3.3. Técnicas de Detección de fallas Basadas en Sistemas Expertos**

Los Sistemas Expertos son algoritmos que incorporan conocimiento sobre un área o dominio particular con el fin de resolver problemas relacionados exclusivamente con esa área. Los sistemas expertos usualmente constan de dos componentes principales; una base de conocimientos y un mecanismo de inferencia. La base de conocimientos contiene la experiencia específica requerida en el área en la que se desempeñará el sistema experto, la cual se puede expresar mediante reglas SI-ENTONCES (IF-THEN), proposiciones, marcos, objetos, procedimientos y casos [45].

Un sistema experto para la detección y diagnóstico de fallas es un sistema con una base de conocimientos que almacena la experiencia acumulada de expertos en diagnóstico de fallas particulares al proceso sobre el que se implementará dicho sistema. Sin embargo, los sistemas expertos no pueden diagnosticar fallas no relacionadas en la base de conocimientos, por lo que no pueden adaptarse a nuevos modos de operación o a nuevas condiciones de falla de los procesos.

J.C. da Silva et al [52] presentan un sistema experto que usa una combinación de modelamiento orientado a objetos, reglas definidas y redes semánticas para detectar las fallas más comunes en sensores, tales como envejecimiento y descalibración, así como también para detectar fallas del proceso mismo. En [52] se describe también el desarrollo un módulo de corrección de sensores basado en la extracción de parámetros de falla (para cada uno de los modos de falla del sensor).

J. Montmain y S. Gentil [54] desarrollan un sistema experto de supervisión y diagnóstico para ayudar al operador, el cual está basado en grafos de causalidad. Un grafo de causalidad representa un proceso de manera altamente abstracta y puede adaptarse para modelar el conocimiento de un proceso con distintos grados de precisión en los modelos matemáticos subyacentes a los procesos. Implementan además un sistema de inferencia difuso como interfaz numérico/simbólica entre los indicadores de falla cuantitativos y el razonamiento de diagnóstico simbólico en los mismos, la cual además es una herramienta de toma de decisiones efectiva en entornos imprecisos o con mucha incertidumbre. Se presenta también en [54] una aplicación del sistema desarrollado en la industria de re-procesamiento de combustible nuclear

#### **2.3.4. Detección de Fallas Basada en Técnicas Híbridas de Inteligencia Artificial**

Las técnicas de inteligencia artificial (IA), tales como las redes neuronales artificiales (ANNs), la lógica difusa y los algoritmos genéticos, han sido aplicadas exitosamente en la detección y diagnóstico automáticos de fallas en procesos. Se pueden aprovechar las ventajas de dos o más técnicas de inteligencia artificial mientras se superan sus desventajas si se combinan en una técnica híbrida. Se muestran a continuación algunos ejemplos de técnicas híbridas de inteligencia artificial aplicadas a la detección de fallas.

B. Samanta et al [50] comparan el desempeño de la detección de fallas llevada a cabo en rodamientos usando dos diferentes clasificadores; redes neuronales artificiales (ANNs) y máquinas de soporte vectorial (SVMs). Las señales de vibraciones de máquinas con rodamientos normales y con rodamientos defectuosos se procesan para la extracción de características, las cuales alimentan luego a los clasificadores. Los parámetros de los clasificadores (e.g. número de neuronas en la capa oculta en el caso de las ANNs o la función del kernel en el caso de las SVMs) y las características de alimentación son seleccionadas usando un optimizador basado en algoritmos genéticos.

El sistema de inferencia neuro-difuso adaptativo (ANFIS) es un modelo híbrido que combina la adaptabilidad de las redes neuronales artificiales con el enfoque cualitativo de la lógica difusa. Mediante la utilización de las propiedades matemáticas de las redes neuronales artificiales con el fin de sintonizar los sistemas difusos (basados en reglas) se hace una aproximación a la forma en la que el cerebro humano procesa información. ANFIS además aprovecha el poder de ambos paradigmas mientras supera sus deficiencias simultáneamente. K. Salahshoor et al [44] proponen una técnica híbrida para el monitoreo de una turbina de vapor industrial mediante la fusión de máquinas de soporte vectorial (SVMs) con un clasificador ANFIS, la cual se utiliza para la detección de errores y para tareas de diagnóstico, obteniendo mejores resultados que con las técnicas aplicadas de manera individual.

### 3. DISEÑO DE UNA ESTRATEGIA ADAPTATIVA POR PCA RECURSIVO PONDERADO (WARP)

Como se ha descrito con anterioridad, los procesos reales presentan cambios de varianzas de las variables a lo largo del tiempo, cambiando consecuentemente la estructura estadística subyacente a los mismos. Se presentan además cambios aleatorios sutiles que pueden desencadenar falsas alarmas eventuales. Debido a esto, se diseña una estrategia recursiva basada en Análisis de Componentes Principales (PCA) para reestructuración de los modelos estadísticos de detección frente a la evidencia de falsas alarmas.

Este capítulo se divide básicamente en cinco secciones. En primer lugar se explica el algoritmo convencional de PCA. En la segunda sección se proponen fórmulas para actualizar de manera recursiva la matriz de autovalores y la matriz de autovectores del *training set* con el fin de incorporar nueva información (datos de proceso) disponible, a lo que se le denomina PCA recursivo o RPCA. Una vez desarrolladas las ecuaciones recursivas se introduce, en la tercera sección del capítulo, el concepto de un factor de ponderación o factor de peso  $w$  el cual es parámetro variable cuyo valor es definido por el usuario, proponiéndose de esta manera un segundo set de ecuaciones recursivas ponderadas (Weighted Adaptive Recursive PCA o WARP). En la cuarta sección se presentan fórmulas para la actualización de los umbrales de decisión de los estadísticos  $T^2$  y  $Q$  respectivamente. Se evalúa finalmente, en la quinta sección, la complejidad computacional del algoritmo propuesto, comparándola además con la complejidad de otros algoritmos recursivos basados en PCA.

La actualización recursiva (así como la recursiva ponderada) es llevada a cabo cuando un nuevo grupo de datos de proceso ha causado una alta tasa de falsas alarmas a pesar de no presentarse una condición de falla en el proceso. Para el discernimiento de estos datos causantes de falsas alarmas de condiciones de fallas reales, se le solicita al operador que clasifique dichos datos dentro de una nueva condición de falla (desconocida) o dentro de operación normal del proceso. Esta actualización recursiva evitará falsas alarmas causadas por la variación natural del proceso en el tiempo (envejecimiento) mediante la incorporación de esta nueva información en el modelo de detección, sin embargo, no evita las falsas alarmas causadas por la incertidumbre intrínseca de los métodos basados en la estadística multivariada, tales como PCA.

### 3.1. PCA Convencional

El algoritmo convencional de PCA consta básicamente de 2 etapas; entrenamiento y detección. Se debe llevar a cabo, un pre-tratamiento de datos antes de la fase de entrenamiento, sin embargo, en este documento se ha incluido por practicidad dicho pre-tratamiento de datos dentro de la etapa de entrenamiento. Se muestran a continuación las dos etapas concernientes al algoritmo de PCA.

#### 3.1.1. Entrenamiento

Considérese un conjunto de datos consistente en  $m$  variables de observación y  $n$  observaciones para cada una de las variables, el cual está dispuesto en un arreglo matricial  $X^{(0)} \in \mathbb{R}^{n \times m}$ , tal como se muestra a continuación:

$$X^{(0)} = \begin{bmatrix} x_{11}^{(0)} & x_{12}^{(0)} & \dots & x_{1m}^{(0)} \\ x_{21}^{(0)} & x_{22}^{(0)} & \dots & x_{2m}^{(0)} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1}^{(0)} & x_{n2}^{(0)} & \dots & x_{nm}^{(0)} \end{bmatrix} \quad (3.1)$$

El superíndice (0) denota que al conjunto de datos de entrenamiento o “*training set*” aún no se le ha realizado un pre-tratamiento de datos. Dicho pre-tratamiento de datos consiste en una estandarización de los datos; se subtrae a cada dato la media correspondiente a su variable y se divide luego por la respectiva desviación estándar. Si se define  $b \in \mathbb{R}^m$  como el “*vector de medias*”, i.e., un vector columna que contiene las medias de las variables, éste se puede calcular como sigue:

$$b = \frac{1}{n_0} (X^{(0)})^T I_n = \begin{bmatrix} \bar{x}_1^{(0)} & \bar{x}_2^{(0)} & \dots & \bar{x}_m^{(0)} \end{bmatrix}^T, \quad (I_n = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^n) \quad (3.2)$$

Si se tienen además las desviaciones estándar de las variables en una matriz diagonal:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_m \end{bmatrix} \in \mathbb{R}^{m \times m} \quad (3.3)$$

el pre-tratamiento de datos se puede llevar a cabo como sigue:

$$X = (X^0 - I_n b^T) \Sigma^{-1} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \in \mathbb{R}^{n \times m} \quad (3.4)$$

Se define ahora la **matriz de covarianza**:

$$S = \frac{1}{n-1} X^T X, \quad S \in \mathbb{R}^{m \times m} \quad (3.5)$$

Una descomposición espectral de  $S$  puede obtenerse si se hallan sus valores propios o *autovalores* y sus respectivos vectores propios o *autovectores*. Se pueden encontrar los autovalores,  $\lambda_i$ , de  $S$  resolviendo:

$$\det(\lambda I - S) = 0 \quad \rightarrow \quad \lambda_1, \lambda_2, \dots, \lambda_m \quad (3.6)$$

donde  $I$  es la matriz identidad. Se deben organizar los autovalores de manera que se cumpla:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \quad (3.7)$$

y se encuentran los autovectores,  $\mathbf{v}_i$ , resolviendo:

$$(\lambda_i I - S) \mathbf{v}_i = 0, \quad \mathbf{v}_i \in \mathbb{R}^m, \quad i = 1, 2, \dots, m \quad (3.8)$$

Se define entonces la **matriz de autovalores**,  $\Lambda$ , como sigue:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \lambda_m \end{bmatrix}, \quad \Lambda \in \mathbb{R}^{m \times m} \quad (3.9)$$

y se define también la **matriz de autovectores**:

$$V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_m], \quad V \in \mathbb{R}^{m \times m} \quad (3.10)$$

luego, la descomposición espectral de  $S$  es:

$$S = V \Lambda V^T \quad (3.11)$$

La proyección  $\mathbf{y} = V^T \mathbf{x}$  de un vector de observación  $\mathbf{x} \in \mathbb{R}^m$  convierte el espacio de observación en un conjunto de variables no correlacionadas correspondientes a los

elementos de  $\mathbf{y}$ . La varianza del  $i$ -ésimo elemento de  $\mathbf{y}$  es igual al  $i$ -ésimo autovalor en la matriz  $\Lambda$ . Sabiendo que  $S$  es invertible (ya que es una matriz simétrica) y con la definición:

$$\mathbf{z} = \Lambda^{-1/2} V^T \mathbf{x} \quad (3.12)$$

el estadístico  $T^2$  de Hotelling está dado por:

$$T^2 = \mathbf{z}^T \mathbf{z} \quad (3.13)$$

Con el fin de capturar de manera óptima la variabilidad de los datos y a la vez minimizar el efecto del ruido, se disponen las primeras  $a$  columnas de la matriz  $V$  (llamadas vectores de carga), correspondientes a los primeros  $a$  autovalores, en una matriz  $P \in \mathbb{R}^{m \times a}$  denominada **matriz de carga**:

$$P = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_a] \quad (3.14)$$

$a$  se conoce como el **número de componentes principales**. La proyección de las observaciones, contenidas en  $X$ , en un espacio de dimensión menor estarán contenidas en la **matriz de puntuación**:

$$T = XP \in \mathbb{R}^{n \times a} \quad (3.15)$$

y la proyección de  $T$  de vuelta al espacio de observación  $m$ -dimensional es:

$$\hat{X} = TP^T \quad (3.16)$$

Se define la **matriz de residuales** como sigue:

$$E = X - \hat{X} \quad (3.17)$$

La matriz de residuales captura las variaciones en el espacio de observación contenido en los vectores de carga asociados con los  $m - a$  menores autovalores. Los subespacios definidos por  $\hat{X}$  y  $E$  se conocen como **espacio de puntuación** y **espacio residual**, respectivamente.

Si se define  $\mathbf{t}_i$  como la  $i$ -ésima columna de  $T$ , se cumplen las siguientes propiedades [3]:

- $Var(\mathbf{t}_1) \geq Var(\mathbf{t}_2) \geq \dots \geq Var(\mathbf{t}_a)$ .
- $Media(\mathbf{t}_i) = 0; \forall i$ .



- $\mathbf{t}_i^T \mathbf{t}_k = 0; \forall i \neq k$ .
- No existe otra expansión ortogonal de  $a$  componentes que capture más variación de los datos.

Dado un vector de observación  $\mathbf{x} \in \mathbb{R}^m$ , éste se puede proyectar al espacio de puntuación de menor dimensión:

$$\mathbf{t}_i = \mathbf{x}^T \mathbf{p}_i \quad (3.18)$$

donde  $\mathbf{p}_i$  es el  $i$ -ésimo vector de carga. A  $\mathbf{t}_i$  también se le conoce como el  $i$ -ésimo **componente principal de  $\mathbf{x}$** . El número de componentes principales,  $a$ , depende del porcentaje de la varianza total de los datos [19] que se desea explicar. Si se desea tener un 95% de explicación, por ejemplo, el número de componentes principales se determina resolviendo:

$$\frac{\sum_{j=1}^a \lambda_j}{\sum_{j=1}^m \lambda_j} \geq 0.95 \quad \rightarrow \quad a \quad (3.19)$$

### 3.1.2. Detección de Fallas

Defínase una matriz  $\Gamma \in \mathbb{R}^{m \times m}$ , tal que:

$$\Lambda = \Gamma^T \Gamma \quad (3.20)$$

Esta matriz coincide con la matriz diagonal correspondiente a la descomposición en valores singulares de  $\frac{1}{\sqrt{n-1}}X$ :

$$\frac{1}{\sqrt{n-1}}X = U\Gamma V^T \quad (3.21)$$

Si se define ahora  $\Gamma_a \in \mathbb{R}^{m \times a}$  como la matriz que contiene las  $a$  primeras filas y columnas de  $\Gamma$ , el estadístico  $T^2$  de Hotelling para el espacio de menor dimensión es:

$$T^2 = \mathbf{x}^T P \Gamma_a^{-2} P^T \mathbf{x} \quad (3.22)$$

El umbral de detección para el estadístico  $T^2$  es:

$$T_\alpha^2 = \frac{a(n-1)(n+1)}{n(n-a)} F_\alpha(a, n-a) \quad (3.23)$$

Las observaciones correspondientes a los  $m-a$  menores autovalores puede ser monitoreado mediante el estadístico  $Q$ , también conocido como SPE (Squared Prediction Error). Si se define:

$$\mathbf{r} = (I - PP^T)\mathbf{x} \quad (3.24)$$

entonces el estadístico  $Q$  se calcula como sigue:

$$Q = \mathbf{r}^T \mathbf{r} \quad (3.25)$$

El umbral de detección para el estadístico  $Q$  viene dado por:

$$Q_\alpha = \theta_1 \left[ \frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \quad (3.26)$$

donde:

$$\theta_i = \sum_{j=a+1}^n \sigma_j^{2i} \quad (3.27)$$

$$h_{0k+1} = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \quad (3.28)$$

y  $c_\alpha$  es la desviación normal correspondiente al nivel de confianza  $(1 - \alpha)$ .

### 3.2. Desarrollo de Fórmulas Recursivas para PCA (RPCA)

Sea  $X_k^0 \in \mathbb{R}^{n_k \times m}$  el  $k$ -ésimo training set, ie., el training set que se ha utilizado para la  $k$ -ésima actualización. Con el fin de llevar a cabo un pre-tratamiento de estos datos, se debe conocer:

$$b_k = \frac{1}{n_k} (X_k^0)^T I_{n_k}, \quad (I_{n_k} = [1 \ 1 \ \dots \ 1]^T \in \mathbb{R}^{n_k}) \quad (3.29)$$

que contiene las medias de las  $m$  variables, y:

$$\Sigma_k = \begin{bmatrix} \sigma_1^{(k)} & 0 & \dots & 0 \\ 0 & \sigma_2^{(k)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_m^{(k)} \end{bmatrix} \quad (3.30)$$

que contiene las desviaciones estándar actuales de las  $m$  variables. El pre-tratamiento de datos que se ha llevado a cabo es:

$$X_k = (X_k^0 - I_{n_k} b_k^T) \Sigma_k^{-1} \quad (3.31)$$

La matriz de covarianza de  $X_k$  es calculada como sigue:

$$S_k = \frac{1}{n_k - 1} X_k^T X_k \in \mathbb{R}^{m \times m} \quad (3.32)$$

Se lleva a cabo su descomposición espectral de manera que:

$$S_k = \frac{1}{n_k - 1} X_k^T X_k = V_k \Lambda_k V_k^T \quad (3.33)$$

Ahora, sea  $X_{k+1}^{*0} \in \mathbb{R}^{n_{k+1}^* \times m}$ , con  $n_k \gg n_{k+1}^*$ , un nuevo conjunto de datos (sin pre-tratamiento) el cual ha causado una alta tasa de falsas alarmas a pesar de no pertenecer a una condición atípica de operación. El vector de medias de  $X_{k+1}^{*0}$  se calcula como sigue:

$$b_{k+1}^* = \frac{1}{n_{k+1}^*} (X_{k+1}^{*0})^T I_{n_{k+1}^*} \quad (3.34)$$

Ahora, se define:

$$X_{k+1}^0 = \begin{bmatrix} X_k^0 \\ X_{k+1}^* \\ 0 \end{bmatrix} \in \mathbb{R}^{n_{k+1} \times m}, \quad n_{k+1} = n_k + n_{k+1}^*, \quad n_{k+1} \gg n_{k+1}^* \quad (3.35)$$

como el nuevo training set (sin pre-tratamiento). El vector de medias de  $X_{k+1}^0$  puede ser calculado recursivamente, con un simple promedio ponderado, como se muestra a continuación:

$$b_{k+1} = \frac{n_k}{n_{k+1}} b_k + \frac{n_{k+1} - n_k}{n_{k+1}} b_{k+1}^* \quad (3.36)$$

Definiendo  $\Delta b_{k+1} = b_{k+1} - b_k$ , las desviaciones estándar del nuevo training set se calculan como sigue:

$$\Sigma_{k+1} = \begin{bmatrix} \sigma_1^{(k+1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \sigma_2^{(k+1)} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \sigma_m^{(k+1)} \end{bmatrix} =$$

$$\frac{n_k - 1}{n_{k+1} - 1} \Sigma_k + \frac{n_{k+1} - n_k - 1}{n_{k+1} - 1} \Sigma_{k+1}^* + \frac{n_k}{n_{k+1} - 1} \text{diag} \left( (\Delta b_{k+1} \Delta b_{k+1}^T)(i, i) \right) \quad (3.37)$$

donde  $\text{diag} \left( (\Delta b_{k+1} \Delta b_{k+1}^T)(i, i) \right)$  es una matriz diagonal que contiene exclusivamente los elementos diagonales del producto  $\Delta b_{k+1} \Delta b_{k+1}^T$ .

El pre-tratamiento de datos para  $X_{k+1}^{*0}$  es llevado a cabo usando las medias actualizadas contenidas en  $b_{k+1}$  y así mismo las desviaciones estándar contenidas en  $\Sigma_{k+1}$ :

$$X_{k+1}^* = (X_{k+1}^{*0} - I_{n_{k+1}} b_{k+1}^T) \Sigma_{k+1}^{-1} \quad (3.38)$$

Su matriz de covarianza es:

$$S_{k+1}^* = \frac{1}{n_{k+1}^* - 1} X_{k+1}^{*T} X_{k+1}^* \quad (3.39)$$

Debido a que el pre-tratamiento de datos para fue llevado a cabo con las medias y desviaciones estándar originales, debe ser actualizado. De (3.31) es evidente que:

$$X_k^0 = X_k \Sigma_k + I_{n_k} b_k \quad (3.40)$$

Considerando lo anterior, luego del pre-tratamiento para  $X_{k+1}^0$  se tiene:

$$X_{k+1} = \begin{bmatrix} (X_k \Sigma_k + I_{n_k} b_k - I_{n_k} b_{k+1}) \Sigma_{k+1}^{-1} \\ X_{k+1}^* \end{bmatrix} \quad (3.41)$$

Luego:

$$X_{k+1} = \begin{bmatrix} (X_k \Sigma_k + I_{n_k} \Delta b_{k+1}) \Sigma_{k+1}^{-1} \\ X_{k+1}^* \end{bmatrix} \quad (3.42)$$

Ahora, calculando la matriz de covarianza de  $X_{k+1}$ :

$$S_{k+1} = \frac{1}{n_{k+1} - 1} \begin{bmatrix} ((X_k \Sigma_k + I_{n_k} \Delta b_{k+1}) \Sigma_{k+1}^{-1})^T & X_{k+1}^{*T} \end{bmatrix} \begin{bmatrix} (X_k \Sigma_k + I_{n_k} \Delta b_{k+1}) \Sigma_{k+1}^{-1} \\ X_{k+1}^* \end{bmatrix} \quad (3.43)$$

Reorganizando la expresión anterior:

$$S_{k+1} = \frac{1}{n_{k+1} - 1} \begin{bmatrix} (\Sigma_{k+1}^{-1})^T (\Sigma_k^T X_k^T + \Delta b_{k+1}^T I_{n_k}) & X_{k+1}^{*T} \end{bmatrix} \begin{bmatrix} (X_k \Sigma_k + I_{n_k} \Delta b_{k+1}) \Sigma_{k+1}^{-1} \\ X_{k+1}^* \end{bmatrix} \quad (3.44)$$

y mediante reducción algebraica:

$$S_{k+1} = \frac{n_k - 1}{n_{k+1} - 1} \Sigma_{k+1}^{-1} \Sigma_k S_k \Sigma_k \Sigma_{k+1}^{-1} + \frac{n_{k+1} - n_k - 1}{n_{k+1} - 1} S_{k+1}^* + \frac{n_k}{n_{k+1} - 1} \Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \quad (3.45)$$

El término de arriba es una fórmula recursiva para obtener la matriz de covarianza actualizada,  $S_{k+1}$ , en función de  $S_k$  y  $S_{k+1}^*$ . Nótese que en esta expresión existen dos compensaciones, i.e.:

- La primera compensación es aplicada debido al cambio en las medias, la cual está relacionada con el término  $\Sigma_1^{-1} \Delta b_1 \Delta b_1^T \Sigma_1^{-1}$ .
- La segunda compensación es aplicada debido al cambio en las desviaciones estándar. Se aplica a  $S_k$  multiplicándole por izquierda por  $\Sigma_{k+1}^{-1} \Sigma_k$  y por derecha por  $\Sigma_k \Sigma_{k+1}^{-1}$ .

El papel de la actualización de las desviaciones estándar es diferente del de la actualización de las medias [20]. Si el cambio en las medias no es compensado, las direcciones de los componentes principales podrían ser muy diferentes de las direcciones reales. Por otro lado, la compensación de las desviaciones estándar afecta únicamente los pesos relativos de cada variable. Las desviaciones estándar no cambian drásticamente si el proceso muestra un cambio suave en el tiempo (envejecimiento), de otra forma se trataría de una falla. Además, el tamaño,  $n_{k+1}^*$ , del nuevo grupo de datos es mucho menor que  $n_{k+1}$ , y por lo tanto su peso es

despreciable comparado con el peso del training set original. Entonces, se puede omitir la compensación debida al cambio en las desviaciones estándar y aún tener una buena aproximación para la matriz de covarianza:

$$S_{k+1} \approx \frac{n_k - 1}{n_{k+1} - 1} S_k + \frac{n_{k+1} - n_k - 1}{n_{k+1} - 1} S_{k+1}^* + \frac{n_k}{n_{k+1} - 1} \Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \quad (3.46)$$

De la ecuación (3.33) se sabe que  $S_k = \frac{1}{n_k - 1} X_k^T X_k = V_k \Lambda_k V_k^T$ , se define entonces:

$$\tilde{S}_k = \frac{n_k - 1}{n_{k+1} - 1} S_k = V_k \left( \frac{n_k - 1}{n_{k+1} - 1} \Lambda_k \right) V_k^T = V_k \tilde{\Lambda}_k V_k^T \quad (3.47)$$

con:

$$\tilde{\Lambda}_k \equiv \frac{n_k - 1}{n_{k+1} - 1} \Lambda_k \quad (3.48)$$

Remplazando las ecuaciones (3.39) y (3.47) en (3.46):

$$S_{k+1} = \tilde{S}_k + \frac{1}{n_{k+1} - 1} X_{k+1}^{*T} X_{k+1}^* + \frac{n_k}{n_{k+1} - 1} \Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \quad (3.49)$$

Reorganizando se tiene que:

$$S_{k+1} = \tilde{S}_k + \frac{1}{n_{k+1} - 1} \left( X_{k+1}^{*T} X_{k+1}^* + (\sqrt{n_k} \Sigma_{k+1}^{-1} \Delta b_{k+1}) (\sqrt{n_k} \Delta b_{k+1}^T \Sigma_{k+1}^{-1}) \right) \quad (3.50)$$

luego:

$$S_{k+1} = \tilde{S}_k + \frac{1}{n_{k+1} - 1} \begin{bmatrix} X_{k+1}^{*T} & \sqrt{n_k} \Sigma_{k+1}^{-1} \Delta b_{k+1} \end{bmatrix} \begin{bmatrix} X_{k+1}^* \\ \sqrt{n_k} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \end{bmatrix} \quad (3.51)$$

Definiendo:

$$\alpha_{k+1} = \frac{1}{n_{k+1} - 1} \quad (3.52)$$

y

$$\Omega_{k+1} = [X_{k+1}^{*T} \quad \sqrt{n_k} \Sigma_{k+1}^{-1} \Delta b_{k+1}] \rightarrow \Omega_{k+1}^T = \begin{bmatrix} X_{k+1}^* \\ \sqrt{n_k} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \end{bmatrix} \quad (3.53)$$

la ecuación (3.51) se convierte en:

$$S_{k+1} = \tilde{S}_k + \alpha_{k+1} \Omega_{k+1} \Omega_{k+1}^T \quad (3.54)$$

Con esta forma,  $S_{k+1}$  es una modificación de *rango 1* de  $\tilde{S}_k$ . Ahora, aplicando siguiente teorema [36]: Si,

$$\tilde{S}_k = V_k \tilde{\Lambda}_k V_k^T \quad (3.55)$$

luego:

$$\begin{aligned} S_{k+1} &= \tilde{S}_k + \alpha_{k+1} \Omega_{k+1} \Omega_{k+1}^T = V_k (\tilde{\Lambda}_k + \alpha_{k+1} z_{k+1} z_{k+1}^T) V_k^T, \\ z_{k+1} &= V_k^T \Omega_{k+1} \end{aligned} \quad (3.56)$$

Entonces, la descomposición espectral de  $\tilde{\Lambda}_k + \alpha_{k+1} z_{k+1} z_{k+1}^T$  debe ser calculada como sigue:

$$\tilde{\Lambda}_k + \alpha_{k+1} z_{k+1} z_{k+1}^T = R_{k+1} D_{k+1} R_{k+1}^T \quad (3.57)$$

Luego, remplazando (3.57) en (3.56):

$$S_{k+1} = V_k (R_{k+1} D_{k+1} R_{k+1}^T) V_k^T \quad (3.58)$$

Reagrupando la expresión de arriba:

$$S_{k+1} = (V_k R_{k+1}) D_{k+1} (R_{k+1}^T V_k^T) \quad (3.59)$$

Nótese que la ecuación (3.59) es de hecho la descomposición espectral de  $S_{k+1}$ , entonces:

$$\mathbf{V}_{k+1} = \mathbf{V}_k \mathbf{R}_{k+1} \quad (3.60)$$

y

$$\mathbf{\Lambda}_{k+1} = \mathbf{D}_{k+1} \quad (3.61)$$

son, respectivamente, las fórmulas recursivas para la matriz de autovectores y la matriz de autovalores.

Se han planteado, hasta ahora, fórmulas recursivas que permiten actualizar fácilmente las medias y desviaciones estándar de las variables, la matriz de covarianza y las matrices de autovalores y autovectores con el fin de llevar a cabo un re-entrenamiento. Se plantearán a continuación fórmulas recursivas con la introducción de un factor de ponderación que asigna pesos tanto a los datos previos como a los nuevos.

### 3.3. Desarrollo de Fórmulas Recursivas Ponderadas para PCA (WARP)

Considérese que la cantidad de nuevos datos sea considerablemente menor que la cantidad de datos en la etapa previa de entrenamiento. Esto podría causar que los nuevos datos no tenga un impacto considerable en la actualización recursiva del modelo estadístico de detección de fallas.

Lo que se ha establecido en el párrafo anterior resulta evidente si se considera la ecuación (3.45), en la cual se puede observar que el peso para  $S_k$  es el cociente entre sus grados de libertad y la cantidad total de grados de libertad disponibles;  $\frac{n_k-1}{n_{k+1}-1}$ , y el peso de  $S_{k+1}^*$  es también el cociente entre sus grados de libertad y la cantidad total de grados de libertad;  $\frac{n_{k+1}-n_k-1}{n_{k+1}-1}$ . Si  $n_{k+1} \gg n_{k+1}^*$ , entonces:

$$\frac{n_k - 1}{n_{k+1} - 1} \gg \frac{n_{k+1} - n_k - 1}{n_{k+1} - 1} \quad (3.62)$$

El peso del término restante,  $\Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1}$ , es  $\frac{n_k}{n_{k+1}-1}$  y difiere del peso de  $S_k$  sólo por la cantidad  $\frac{1}{n_{k+1}-1}$ , la cual tiende a cero si  $n_{k+1}$  es muy grande.

Se podría definir entonces un factor de ponderación o factor de aprendizaje,  $w$ , el cual está relacionado con cuánta importancia o “*peso*” se quiere asignar a los nuevos datos comparada con el *peso* de los datos previos, de manera que la ecuación (3.54) se convierte en:

$$S_{k+1} = (1 - w) \Sigma_{k+1}^{-1} \Sigma_k S_k \Sigma_k \Sigma_{k+1}^{-1} + w S_{k+1}^* + (1 - w) \Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1}, \quad 0 \leq w \leq 1 \quad (3.63)$$



la fórmula recursiva para la actualización de las medias se convierte en:

$$\mathbf{b}_{k+1} = (1 - w)\mathbf{b}_k + w \mathbf{b}_{k+1}^* \quad (3.64)$$

y la fórmula recursiva para la actualización de las desviaciones estándar se convierte en:

$$\Sigma_{k+1} = (1 - w)\Sigma_k + w \Sigma_{k+1}^* + (1 - w)\text{diag}\left((\Delta \mathbf{b}_{k+1} \Delta \mathbf{b}_{k+1}^T)(i, i)\right) \quad (3.65)$$

Si la compensación debida al cambio de las desviaciones estándar no se tiene en cuenta, la ecuación (3.63) se convierte en:

$$S_{k+1} = (1 - w)S_k + wS_{k+1}^* + (1 - w)\Sigma_{k+1}^{-1} \Delta \mathbf{b}_{k+1} \Delta \mathbf{b}_{k+1}^T \Sigma_{k+1}^{-1} \quad (3.66)$$

Definiendo:

$$\tilde{S}_k = (1 - w)S_k = V_k((1 - w)\Lambda_k)V_k^T = V_k \tilde{\Lambda}_k V_k^T \quad (3.67)$$

donde:

$$\tilde{\Lambda}_k \equiv (1 - w)\Lambda_k \quad (3.68)$$

se tiene que:

$$S_{k+1} = \tilde{S}_k + \frac{w}{n_{k+1}^* - 1} X_{k+1}^{*T} X_{k+1}^* + (1 - w)\Sigma_{k+1}^{-1} \Delta \mathbf{b}_{k+1} \Delta \mathbf{b}_{k+1}^T \Sigma_{k+1}^{-1} \quad (3.69)$$

Reorganizando:

$$\begin{aligned} & S_{k+1} \\ &= \tilde{S}_k + \left[ \sqrt{\frac{w}{n_{k+1}^* - 1}} X_{k+1}^{*T} \quad \sqrt{1 - w} \Sigma_{k+1}^{-1} \Delta \mathbf{b}_{k+1} \right] \begin{bmatrix} \sqrt{\frac{w}{n_{k+1}^* - 1}} X_{k+1}^* \\ \sqrt{1 - w} \Delta \mathbf{b}_{k+1}^T \Sigma_{k+1}^{-1} \end{bmatrix} \end{aligned} \quad (3.70)$$

Definiendo ahora:

$$\begin{aligned}\Omega_{k+1} &= \begin{bmatrix} \sqrt{\frac{w}{n_{k+1}^* - 1}} X_{k+1}^{*T} & \sqrt{1-w} \Sigma_{k+1}^{-1} \Delta b_{k+1} \end{bmatrix} \rightarrow \Omega_{k+1}^T \\ &= \begin{bmatrix} \sqrt{\frac{w}{n_{k+1}^* - 1}} X_{k+1}^* \\ \sqrt{1-w} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} \end{bmatrix} \quad (3.71)\end{aligned}$$

Entonces:

$$S_{k+1} = \tilde{S}_k + \Omega_{k+1} \Omega_{k+1}^T \quad (3.72)$$

La anterior es una modificación de *rango 1* de  $\tilde{S}_k$ , luego:

$$\tilde{S}_k = V_k \tilde{\Lambda}_k V_k^T \quad (3.73)$$

$$\rightarrow S_{k+1} = \tilde{S}_k + \Omega_{k+1} \Omega_{k+1}^T = V_k (\tilde{\Lambda}_k + z_{k+1} z_{k+1}^T) V_k^T \quad (3.74)$$

con:

$$\Omega_{k+1} = V_k z_{k+1} \rightarrow z_{k+1} = V_k^T \Omega_{k+1} \quad (3.75)$$

Consecuentemente, la descomposición espectral de  $\tilde{\Lambda}_k + z_{k+1} z_{k+1}^T$  debe ser llevada a cabo, como se muestra a continuación:

$$\tilde{\Lambda}_k + z_{k+1} z_{k+1}^T = R_{k+1} D_{k+1} R_{k+1}^T \quad (3.76)$$

Por lo tanto:

$$S_{k+1} = V_k (R_{k+1} D_{k+1} R_{k+1}^T) V_k^T \quad (3.77)$$

Reagrupando la expresión anterior:

$$S_{k+1} = (V_k R_{k+1}) D_{k+1} (R_{k+1}^T V_k^T) \quad (3.78)$$

Nótese que la ecuación (3.78) es de hecho la descomposición espectral de  $S_{k+1}$ , entonces, se tiene:

$$V_{k+1} = V_k R_{k+1} \quad (3.79)$$

y

$$\Lambda_{k+1} = D_{k+1} \quad (3.80)$$

Las ecuaciones (3.79) y (3.80) son las fórmulas recursivas para obtener la matriz de autovectores y la matriz de autovalores respectivamente.

### 3.4. Actualización de Umbrales de Detección

Hasta el momento, se han planteado fórmulas recursivas para la actualización del modelo estadístico de detección de fallas, sin embargo, debido a que se han considerado nuevas muestras de datos para el entrenamiento, hay más grados de libertad, ya que:

$$n_{k+1} - 1 > n_k - 1 \quad (3.81)$$

Consecuentemente, los umbrales de detección deben ser actualizados también. Antes de actualizar los umbrales, el número de componentes principales debe ser recalculado:

$$\frac{\sum_{j=1}^{a_{k+1}} \lambda_j}{\sum_{j=1}^m \lambda_j} \geq 0.95 \quad \rightarrow \quad a_{k+1} \quad (3.82)$$

Los nuevos umbrales se calculan como sigue:

$$T_{\alpha_{k+1}}^2 = \frac{a_{k+1}(n_{k+1} - 1)(n_{k+1} + 1)}{n_{k+1}(n_{k+1} - a_{k+1})} F_{\alpha}(a_{k+1}, n_{k+1} - a_{k+1}) \quad (3.83)$$

y

$$Q_{\alpha_{k+1}} = \theta_{1_{k+1}} \left[ \frac{h_{0_{k+1}} c_{\alpha} \sqrt{2\theta_{2_{k+1}}}}{\theta_{1_{k+1}}} + 1 + \frac{\theta_{2_{k+1}} h_{0_{k+1}} (h_{0_{k+1}} - 1)}{\theta_{1_{k+1}}^2} \right]^{1/h_{0_{k+1}}} \quad (3.84)$$

donde

$$\theta_{i_{k+1}} = \sum_{j=a_{k+1}+1}^{n_{k+1}} \sigma_{j_{k+1}}^{2i} \quad (3.85)$$

$$h_{0_{k+1}} = 1 - \frac{2\theta_{1_{k+1}}\theta_{3_{k+1}}}{3\theta_{2_{k+1}}^2} \quad (3.86)$$

y  $c_\alpha$  es la desviación normal correspondiente al nivel de confianza  $(1 - \alpha)$ . Una vez que las fórmulas recursivas han sido aplicadas y los umbrales han sido actualizados, los modelos estadísticos de detección se convierten en:

$$T^2 = \mathbf{x}^T P_{k+1} \Gamma_{a_{k+1}}^{-2} P_{k+1}^T \mathbf{x} \leq T_{\alpha_{k+1}}^2 \quad (3.87)$$

$$Q = \mathbf{r}^T \mathbf{r} \leq Q_{\alpha_{k+1}}, \quad \mathbf{r} = (1 - P_{k+1} P_{k+1}^T) \mathbf{x} \quad (3.88)$$

### 3.5. Complejidad Computacional

Se comparará la complejidad computacional de la actualización recursiva con la complejidad computacional de un re-entrenamiento no recursivo, i.e., un entrenamiento con PCA convencional usando  $X_{k+1}$  como *training set*. En esta comparación se incluyen además dos métodos de PCA recursivo encontrados en la literatura, los cuales son propuestos por Weihua L. et al [20] y J.C. Jeng [5]. Se muestra la comparación de complejidad computacional en la *Tabla 1*.

Como criterio para la comparación de la complejidad computacional se ha usado el número de puntos de operación de coma flotante (FLOPS). Para el caso del algoritmo propuesto en [5] la actualización recursiva debe llevarse a cabo cada instante de muestreo, por lo que las expresiones que representan el número de FLOPs son para cada instante, consecuentemente cada término debe ser multiplicado por  $n_{k+1}^*$  si se van a evaluar  $n_{k+1}^*$  instantes de muestreo.

Se graficará, para tres casos, el número de FLOPS requerido para llevar a cabo la actualización usando los métodos recursivos y para PCA convencional. No se graficará el número de FLOPs usados para la actualización llevada a cabo por el algoritmo desarrollado en [5] ya que su complejidad computacional crece muy rápido con  $n_{k+1}^*$  y la escala no permite una visualización adecuada de las curvas para los otros métodos.

**Tabla 1 - Comparación de la complejidad computacional de actualización recursiva vs actualización no recursiva.**

Algoritmo	Weihoa L., H. Yue, S. Valle-Cervantes, S.J. Qin [20].	J.C. Jeng [5].	I. Portnoy, M. Sanjuán.	PCA Convencional [3].
Actualización de Medias	$3mn_{k+1} + 2m$	$3mn_{k+1}$	$4mn_{k+1}^* + 2m$	$4mn_{k+1} - m$
Actualización de desviaciones estándar	$4mn_{k+1}^* + 6m$	Not Applicable	$5m^2$	$2mn_{k+1} + 2m^2$
Actualización de Matriz de Covarianza	$2m^2n_{k+1} + 2m^2n_{k+1}^* + 10m^3 + 4m^2 + m$	$2m^2n_{k+1}^* + 2m^2 + 2m - 1$	$2m^2n_{k+1}^* + 10m^3 + 3m^2 + m$	$4m^2n_{k+1} + 4mn_{k+1} - m^2 + m$
Descomposiciones Espectrales	$O(2m^3)$	$O(m^3)$	$O(m^3)$	$O(m^3)$
Actualización de Matriz de Autovalores	$m^2(2m - 1)$	$m^2(2m - 1)$	$m^2(2m - 1)$	$m^2$
Actualización de Matriz de Autovectores	0	$m^2$	$m^2$	$m^2$

Los casos en los cuales se llevará a cabo la comparación de complejidad computacional son:

- Caso 1:  $n_{k+1} = 20000$ ,  $n_{k+1}^* = 1000$ ,  $m = 10, 11, \dots, 100$
- Caso 2:  $n_{k+1} = 100000$ ,  $n_{k+1}^* = 5000$ ,  $m = 10, 11, \dots, 100$
- Caso 3:  $n_{k+1} = 10^5$ ,  $n_{k+1}^* = 5 \times 10^3$ ,  $m = 10, 11, \dots, 100$

Se muestran los resultados de los Casos 1, 2 y 3 en la *Figura 7*, *Figura 8* y *Figura 9* respectivamente.

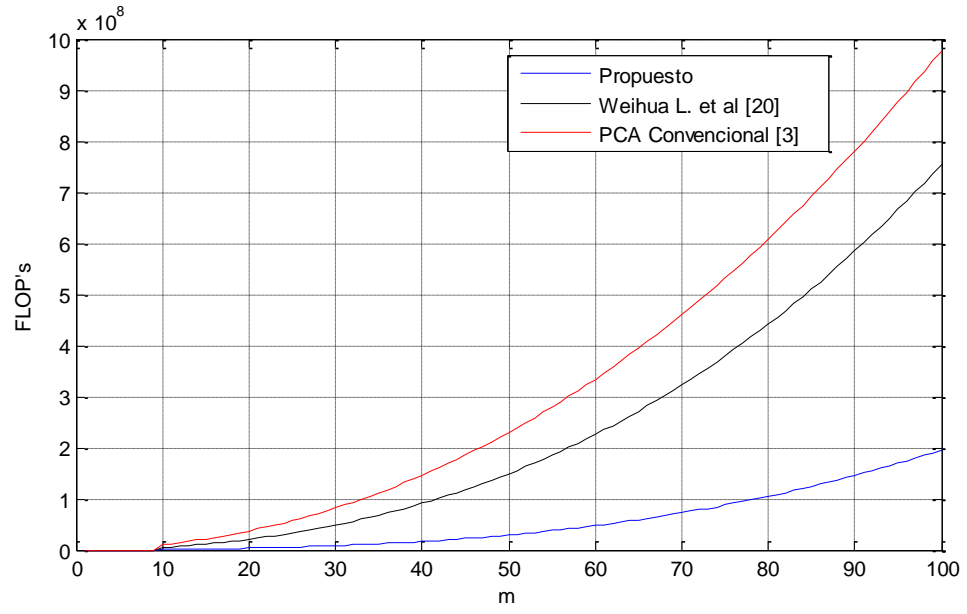


Figura 7 – Evaluación de complejidad computacional, caso 1:  $n_{k+1} = 20000$ ,  $n_{k+1}^* = 1000$ .

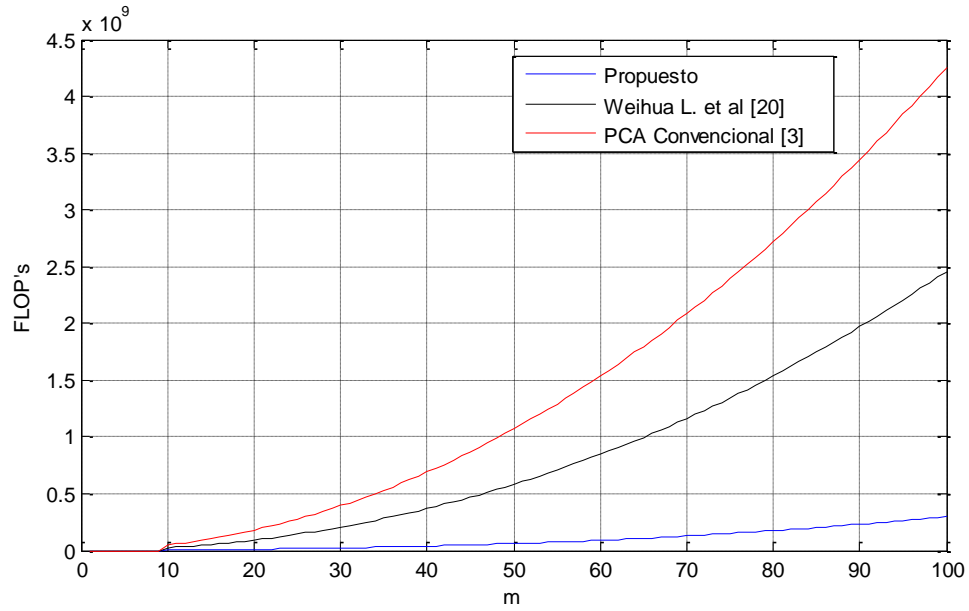
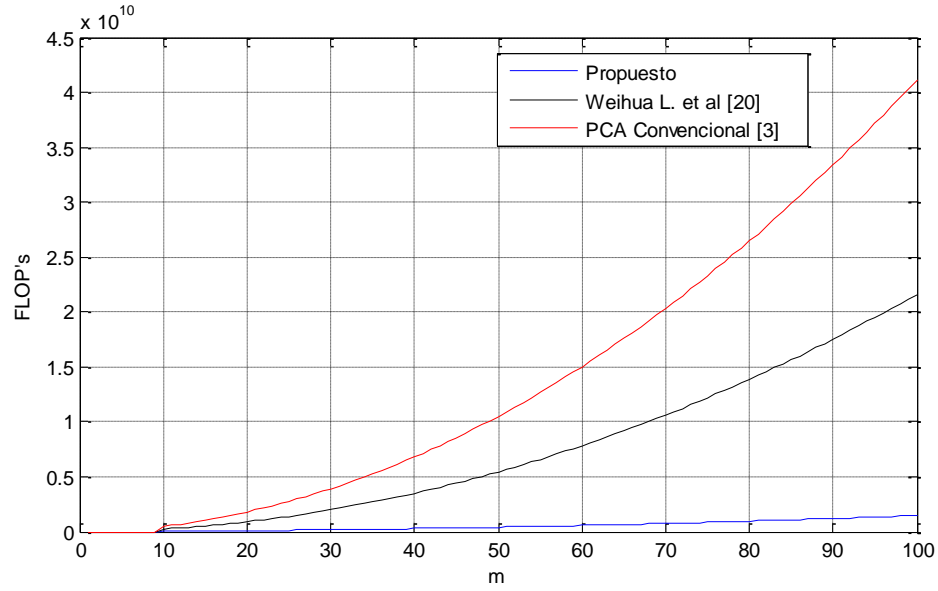


Figura 8 - Evaluación de complejidad computacional, caso 2:  $n_{k+1} = 100000$ ,  $n_{k+1}^* = 5000$ .



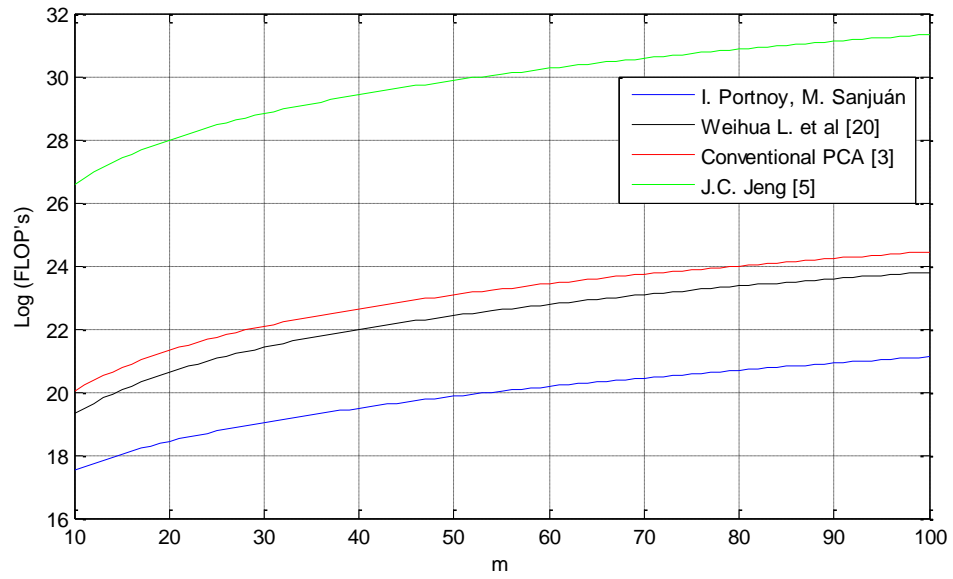
**Figura 9 - Evaluación de complejidad computacional, caso 3:  $n_{k+1} = 1000000$ ,  $n_{k+1}^* = 50000$ .**

Es evidente, de la *Figura 7*, *Figura 8* y *Figura 9*, que mientras más datos se tengan, mayor se hace la diferencia entre la complejidad computacional entre una actualización recursiva y una actualización no recursiva. Por lo tanto, las fórmulas recursivas reducen significativamente la demanda computacional.

El algoritmo propuesto por J.C. Jeng [5] está diseñado exclusivamente para llevar a cabo una actualización por cada muestreo, debido a lo cual requerirá un número mucho mayor de operaciones de coma flotante si se le compara con los otros algoritmos durante  $n_{k+1}^*$  instantes de muestro. Para poder hacer una comparación cuantitativa de la complejidad computacional que incluya a la técnica desarrollada en [5] se utilizará una escala logarítmica para los FLOP's. Se hace dicha comparación para:

$$n_{k+1} = 10^5, n_{k+1}^* = 5 \times 10^3, m = 10, 11, \dots, 100$$

condiciones que corresponden al *Caso 3*. Los resultados de la comparación se muestran en la *Figura 10*.



**Figura 10 – Evaluación de complejidad computacional incluyendo algoritmo de [5], caso 3:  $n_{k+1} = 1000000$ ,  $n_{k+1}^* = 50000$ , escala logarítmica.**

Se evidencia entonces que el algoritmo propuesto lleva a cabo la actualización recursiva con menor complejidad computacional respecto a los otros algoritmos con los que se ha comparado.



## 4. DISEÑO DE LA ESTRATEGIA ADAPTATIVA CON UMBRAL DINÁMICO: MOVING AVERAGE FILTER PCA (MAF-PCA)

Con el enfoque de WARP las falsas alarmas causadas por los cambios del proceso en el tiempo son evitadas. Sin embargo, las falsas alarmas causadas por variaciones leves del proceso (aquellas que no cambian su estructura estadística) y por la incertidumbre natural de los métodos estadísticos, como es el caso de PCA, deben ser evitadas también. Una ventana móvil o media móvil será implementada directamente al estadístico  $T^2$ , así como también para el estadístico  $Q$ , en lugar de implementarlo a los datos de proceso. Además de la ventana móvil, se diseña, para ambos casos, un algoritmo de filtro que decida cuál será el umbral para cada instante de muestro basándose en una observación de la media y desviación estándar de la ventana.

### 4.1. Estrategia Adaptativa con Umbral Dinámico para Estadístico $T^2$

Sea  $H_T \in \mathbb{R}^k$  la ventana móvil que contiene los últimos  $k$  valores del estadístico  $T^2$ :

$$H_T = \begin{bmatrix} T_{i-k+1}^2 \\ \vdots \\ T_{i-1}^2 \\ T_i^2 \end{bmatrix} \quad (4.1)$$

La media,  $\bar{H}_T$ , y la desviación estándar,  $\sigma(H_T)$ , de la ventana móvil deben ser calculadas. Ahora, considérense los  $r$  últimos valores del estadístico  $T^2$ :  $T_i^2, T_{i-1}^2, \dots, T_{i-r+1}^2$ , con  $r < k$ . Si todos los últimos  $r$  valores consecutivos de  $T^2$  son mayores que el estadístico de prueba  $T_\alpha^2$  se dice que hay una *racha*. Es evidente además que:

$$k \in \mathbb{Z}^+ \quad \wedge \quad r \in \mathbb{Z}^+ \quad (4.2)$$

Las reglas de decisión para el ajuste del umbral, para el estadístico  $T^2$  se definen a continuación:

- Si  $T_i^2 > T_\alpha^2 \wedge T_{i-1}^2 > T_\alpha^2 \wedge \dots \wedge T_{i-r+1}^2 > T_\alpha^2$ , entonces el umbral será  $T_\alpha^2$  y una alarma será desencadenada.
- Si  $T_i^2 < T_\alpha^2 \vee T_{i-1}^2 < T_\alpha^2 \vee \dots \vee T_{i-r+1}^2 < T_\alpha^2$ , entonces:
  - Si  $T_\alpha^2 \geq \bar{H}_T + \sigma(H_T)$ , entonces el umbral será  $T_\alpha^2$ .
  - Si  $T_\alpha^2 < \bar{H}_T + \sigma(H_T)$ , entonces el umbral será  $\bar{H}_T + \sigma(H_T)$ .

Se hace una ilustración con diagrama de flujo del algoritmo diseñado, con  $r = 3$ , en la *Figura 11*:

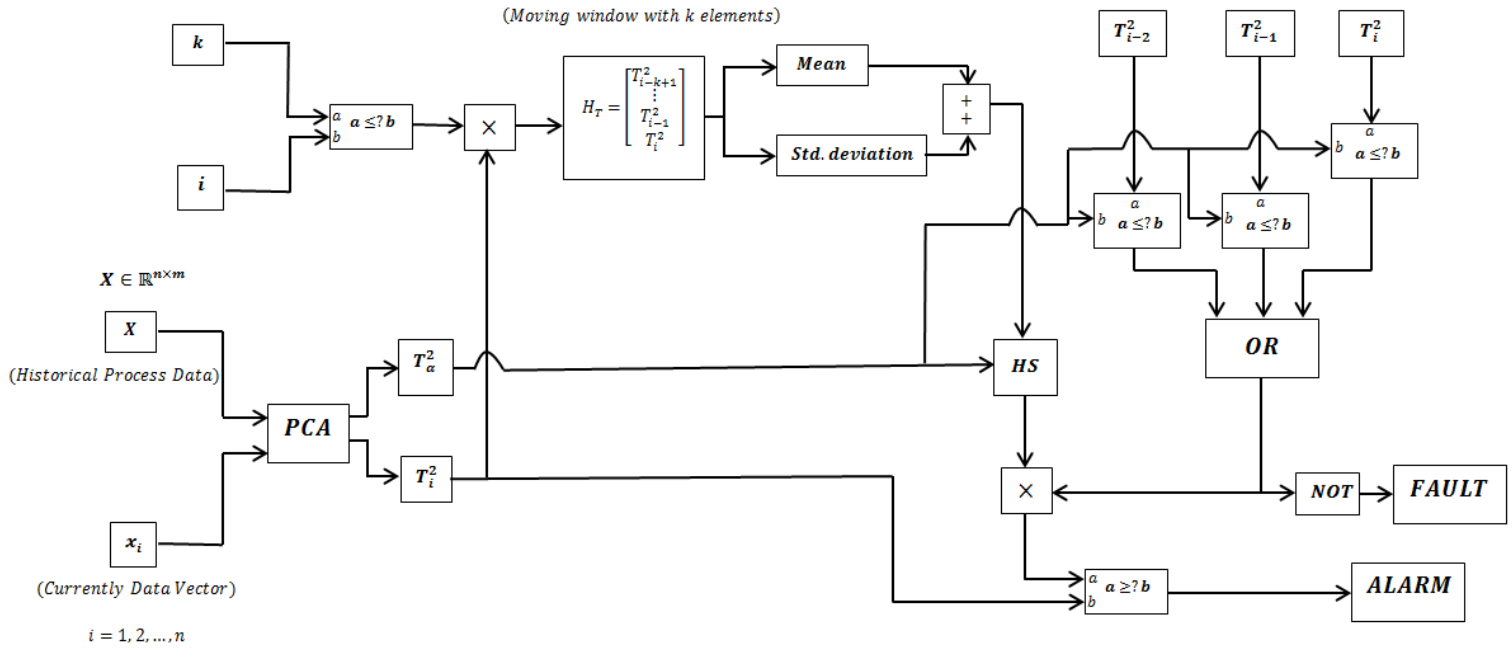


Figura 11 – Esquema del algoritmo MAF-PCA para el estadístico  $T^2$ , con  $r = 3$ .

En la *Figura 11*, el bloque HS hace referencia a un “High Selector”, i.e., un bloque lógico que devuelve como salida la entrada con mayor valor que llegue a él. Nótese que se ha representado una racha de 3 tiempos de muestreo consecutivos en los cuales  $T_i^2 > T_\alpha^2$ , sin embargo el tamaño de la racha,  $r$ , es un parámetro que puede variarse, al igual que el tamaño de la ventana móvil,  $k$ .

Dado que se tienen dos parámetros cuyo valor es variable,  $r$  &  $k$ , es necesario hacer un diseño de experimento (DOE), con el fin de determinar los valores óptimos para

estos parámetros. Este diseño de experimento, sin embargo, podría tener resultados distintos de acuerdo al proceso sobre el cual se aplicó el algoritmo de detección de fallas, por lo que sus resultados no pueden ser generalizados.

El proceso sobre el cual se aplica el algoritmo de detección de fallas es el Tennessee Eastman Process (TEP), el cual ha sido usado ampliamente como proceso de prueba o *benchmark process* para realizar detección de fallas, y el cual se describe con más detalle en el siguiente capítulo.

Se lleva a cabo a continuación un diseño de experimento (DOE). Se definen en primer lugar dos regiones de experimentación:

- Región de experimentación 1:

$$10 \leq k \leq 50, \quad 3 \leq r \leq 9$$

- Región de experimentación 2:

$$50 \leq k \leq 90, \quad 3 \leq r \leq 19$$

El DOE constará de tres etapas; un experimento factorial en la región de experimentación 1, un experimento factorial en la región de experimentación 2 y una optimización. Las primeras dos etapas buscan determinar la significancia de los factores  $r$  &  $k$  en las regiones de experimentación definidas, para la variable de respuesta; la tasa de falsas alarmas. La tercera etapa busca encontrar los valores óptimos de dichos factores.

#### **4.1.1. Experimento Factorial en la Región de Experimentación 1 (Para Estadístico $T^2$ )**

Para llevar a cabo el entrenamiento se utilizan datos del TEP con todas las variables de entrada en estado estable, sin embargo, con el fin de inducir falsas alarmas, para la fase de detección se usan datos generados con perturbaciones de 0,05% respecto a la magnitud de estado estable de las variables manipuladas. Esto se hace también para la región de experimentación 2.

Para este propósito se lleva a cabo un experimento factorial en la región de experimentación 1, i.e.,  $19 \leq k \leq 50$ ,  $3 \leq r \leq 19$ , para los factores y niveles que se definen en la *Tabla 2*. La variable de respuesta será la tasa de falsas alarmas, la cual se define como:

$$Tasa\ de\ F.A. = 100\% \times \left( \sum_{\forall i | T_i^2 > T_{\alpha}^2} 1 \right) / \left( \sum_{\forall i} 1 \right) \quad (4.3)$$

Dicho diseño tiene dos factores y tres niveles para cada factor, tal como se observa en la *Tabla 2*, haciéndose una réplica por condición experimental.

**Tabla 2 - Factores y niveles definidos para la región de experimentación 1 (Para Estadístico  $T^2$ ).**

<b>Experimento Factorial 1 (Para Estadístico <math>T^2</math>)</b>			
<b>Factor</b>	<b>Niveles</b>		
	<b>Bajo</b>	<b>Medio</b>	<b>Alto</b>
Tamaño de Ventana ( $k$ )	10	30	50
Tamaño de Racha ( $r$ )	3	6	9

Los puntos o condiciones experimentales, aleatorizados, se muestran en la *Tabla 3*, junto con los respectivos valores de la variable de respuesta.

**Tabla 3 - Condiciones experimentales de la región de experimentación 1 (Para Estadístico  $T^2$ ).**

<b>Réplica</b>	<b><math>k</math></b>	<b><math>r</math></b>	<b>Tasa de F.A. (%)</b>
1	10	6	5.7692
1	50	6	6.7949
1	50	3	7.3077
1	30	9	8.2051
1	10	9	6.2821
1	50	9	7.1785
1	30	6	6.9872
1	30	3	6.1538
1	10	3	5.8974

2	50	9	7.0513
2	30	3	6.2821
2	30	9	7.2436
2	30	6	6.9872
2	10	3	5.9615
2	50	6	6.5385
2	50	3	7.3077
2	10	6	6.4744
2	10	9	5.2564

Se lleva a cabo un análisis de varianza o ANOVA, con ayuda del software Statgraphics®, cuyo resultado se muestra en la *Tabla 4*.

**Tabla 4 - ANOVA para la región de experimentación 1 (Para Estadístico  $T^2$ ).**

<b>Fuente</b>	<b>Suma de Cuadrados</b>	<b>Grados de Libertad</b>	<b>Cuadrado Medio</b>	<b>F</b>	<b>Valor-P (<math>\alpha = 0.05</math>)</b>
$k$	4.5281	2	2.26405	15.82	0.0011
$r$	0.472594	2	0.236297	1.65	0.2450
$k \cdot r$	2.35425	4	0.588563	4.11	0.0364
Residuales	1.28817	9	0.14313		
Total (Corregido)	8.64312	17			

Se observa en la ANOVA que tanto el factor  $k$  como también la interacción  $k \cdot r$  son significativos en esta región de experimentación con un nivel de confianza del 95%, ya que su valor  $P$  es inferior a 0.05, mientras que el factor  $r$  no es significativo en esta región de experimentación. Se procede a continuación a realizar el experimento factorial en la región de experimentación 2.

#### **4.1.2. Experimento Factorial en la Región de Experimentación 2 (Para Estadístico $T^2$ )**

Al igual que para la región de experimentación 1, en la región de experimentación 2,  $50 \leq k \leq 90$ ,  $3 \leq r \leq 19$ , se lleva a cabo un diseño factorial con dos factores y tres

niveles para cada factor, tal como se observa en la *Tabla 5*, haciéndose un réplica por condición experimental.

**Tabla 5 - Factores y niveles definidos para la región de experimentación 2 (Para Estadístico  $T^2$ ).**

<b>Experimento Factorial 2 (Para Estadístico <math>T^2</math>)</b>			
<b>Factor</b>	<b>Niveles</b>		
	<b>Bajo</b>	<b>Medio</b>	<b>Alto</b>
Tamaño de Ventana ( $k$ )	50	70	90
Tamaño de Racha ( $r$ )	3	11	19

Los puntos o condiciones experimentales, aleatorizados, se obtienen con ayuda del software Statgraphics® y se muestran en la *Tabla 6*.

**Tabla 6 - Condiciones experimentales de la región de experimentación 2 (Para Estadístico  $T^2$ ).**

<b>Réplica</b>	<b><math>k</math></b>	<b><math>r</math></b>	<b>Tasa de F.A. (%)</b>
1	50	11	8.2583
1	90	11	9.4595
1	90	3	10.9610
1	70	19	10.6607
1	50	19	10.0601
1	90	19	10.3604
1	70	11	8.8689
1	70	3	9.0101
1	50	3	7.6577
2	90	19	9.3093
2	70	3	8.5586
2	70	19	8.1081
2	70	11	10.0601
2	50	3	8.4084
2	90	11	9.0090
2	90	3	8.1810
2	50	11	7.3574

2	50	19	8.8589
---	----	----	--------

Para determinar la significancia de los factores se lleva a cabo un análisis de varianza o ANOVA, con ayuda del software Statgraphics®, cuyo resultado se muestra en la *Tabla 7*.

**Tabla 7 - ANOVA para la región de experimentación 2 (Para Estadístico  $T^2$ ).**

<b>Fuente</b>	<b>Suma de Cuadrados</b>	<b>Grados de Libertad</b>	<b>Cuadrado Medio</b>	<b>F</b>	<b>Valor-P (<math>\alpha = 0.05</math>)</b>
$k$	3.91323	2	1.95661	1.76	0.2261
$r$	2.21731	2	1.10866	1.00	0.4060
$k \cdot r$	1.90685	4	0.476712	0.43	0.7844
Residuales	9.9964	9	1.11071		
Total (Corregido)	18.0338	17			

Como se observa, en esta región de experimentación, ninguno de los factores es significativo ya que el valor  $P$  para ambos es mayor que 0.05. Se debe llevar a cabo entonces la optimización dentro de la región de experimentación 1, la cual se hace a continuación.

#### **4.1.3. Optimización (Para Estadístico $T^2$ )**

Luego de haber llevado a cabo los experimentos en las regiones de experimentación 1 y 2, se ha concluido que sólo en la región de experimentación 1 se tiene significancia de al menos un factor. El factor significativo resultó ser  $k$  (tamaño de la ventana móvil), así como también resultó significativa la interacción  $k \cdot r$ . Dado que ya se ha llevado a cabo un experimento con puntos centrales en la región de experimentación 1, se utilizarán los datos de dicho experimento para desarrollar modelos de regresión y evaluar así el punto de operación óptimo.

Se desarrollan distintos modelos de regresión, algunos de los cuales se resumen en la *Tabla 8*.

**Tabla 8 - Resumen de modelos de regresión para optimización de la tasa de falsas alarmas para el estadístico  $T^2$ .**

Modelo	Ajuste ( $R^2$ ajustado)	Significancia del modelo (en ANOVA)	¿Todos los coeficientes significativos?	¿Estructura en los residuales?
$5.05335 + 0.09127k - 0.0012288k^2 + 0.001617k \cdot r$	0.468	OK	NO	SI
$4.50467 + 0.0424167k + 0.00442k \cdot r + 0.0004121k^2 - 0.0002735k^2 \cdot r + 0.0007627k \cdot r^2$	0.584	OK	NO	SI
$4.913485 + 0.1143216k - 0.00152686k^2 + 0.2719261 \sin(k \cdot r)$	0.491	OK	NO	SI
$4.758543 + 0.077675k - 0.0003029k^2 \cdot r + 0.00122346k \cdot r^2 - 0.33364 \sin(k \cdot r)$	0.780	OK	SI	NO

Se implementa finalmente una regresión cuya ecuación se muestra a continuación:

$$F.A.R. = 4.758543 + 0.077675k - 0.0003029k^2 \cdot r + 0.00122346k \cdot r^2 - 0.33364 \sin(k \cdot r) \quad (4.4)$$

En la ecuación (4.4) la variable  $F.A.R.$  hace referencia a la tasa de falsas alarmas (en %). Se muestran las estadísticas de la regresión en la *Tabla 9*.

**Tabla 9 – Estadísticas de la regresión para optimización para estadístico  $T^2$ .**

<b>Estadísticas de la regresión</b>	
Coeficiente de correlación múltiple	0.91212742
Coeficiente de determinación $R^2$	0.83197643
$R^2$ ajustado	0.78027686
Error típico	0.33423251
Observaciones	18

Se muestra el análisis de varianza (ANOVA) para la regresión en la *Tabla 10*.



Tabla 10 - ANOVA de la regresión de la tasa de falsas alarmas para estadístico  $T^2$ .

<i>Fuente</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Cuadrado Medio</i>	<i>F</i>	<i>Valor-P</i> ( $\alpha = 0.05$ )
Regresión	7.19087144	4	1.79771786	11.8234781	$5.91 \times 10^{-5}$
Residuos	1.45224778	13	0.206903364	0.11171137	
Total	8.64311922	17			

Se muestra el análisis de significancia de los coeficientes de regresión en la *Tabla 11*.

Tabla 11 - Análisis de los coeficientes de regresión para optimización para estadístico  $T^2$ .

	<i>Coeficientes</i>	<i>Error típico</i>	<i>Estadístico t</i>	<i>Valor-P</i> ( $\alpha = 0.05$ )
Intercepción	4.758543	0.265937	17.893492	$1.59394 \times 10^{-10}$
K	0.077675	0.012700	6.115766	$3.6854 \times 10^{-5}$
$k^2 \cdot r$	-0.0003029	0.000057	-5.277162	0,00014966
$k \cdot r^2$	0.00122346	0.000218	5.601124	$8.6087 \times 10^{-5}$
$\sin(k \cdot r)$	-0.33364	0,135719	-3.195921	0,00702

Los residuales se muestran en la *Figura 12*. Se deben analizar los residuales para verificar que no tengan una estructura o tendencia evidente.

No se observa una estructura en los residuales, por lo que el modelo de regresión explica bien los resultados.

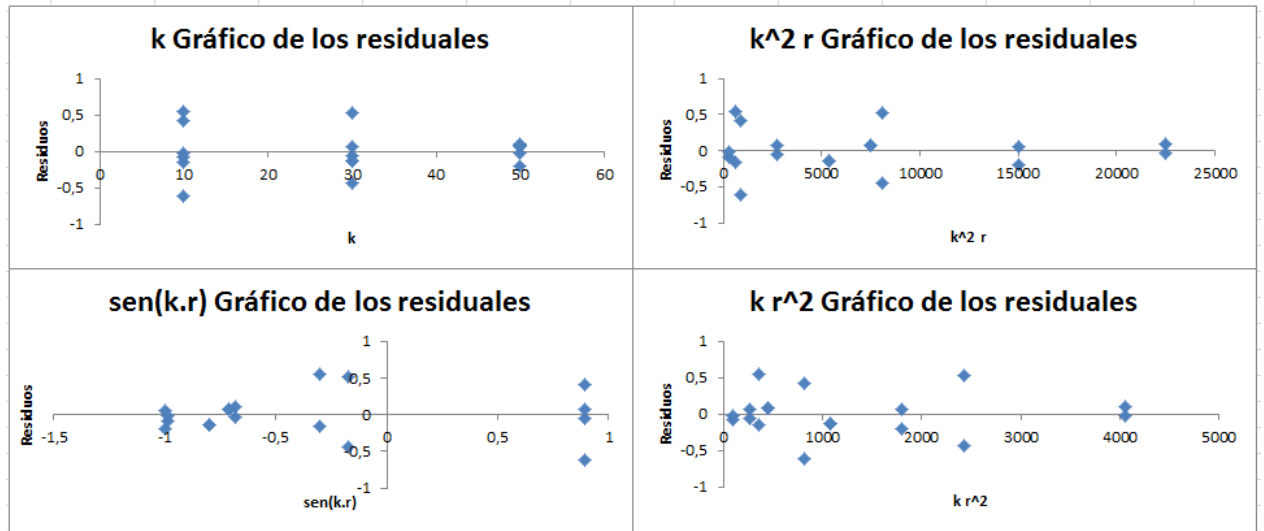
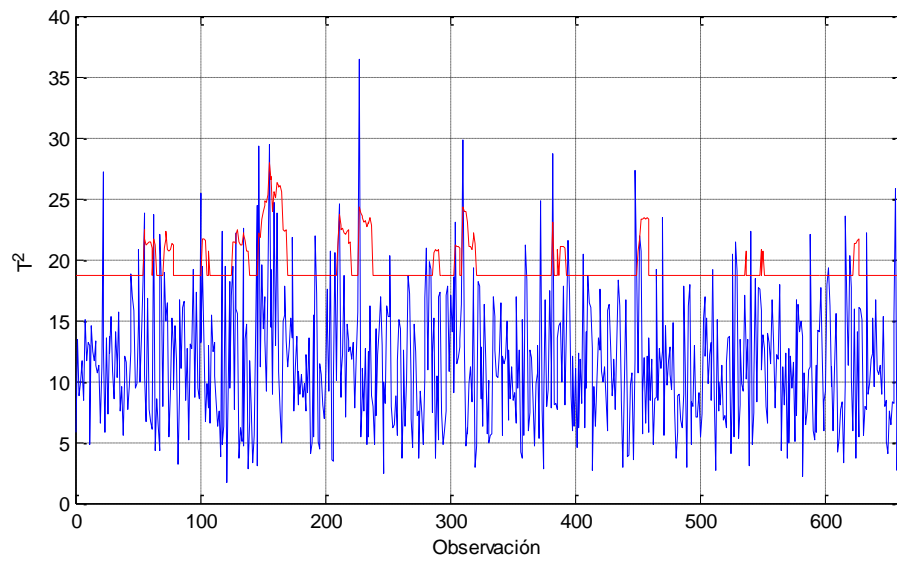


Figura 12 - Gráficos de residuales para regresión de optimización para estadístico  $T^2$ .

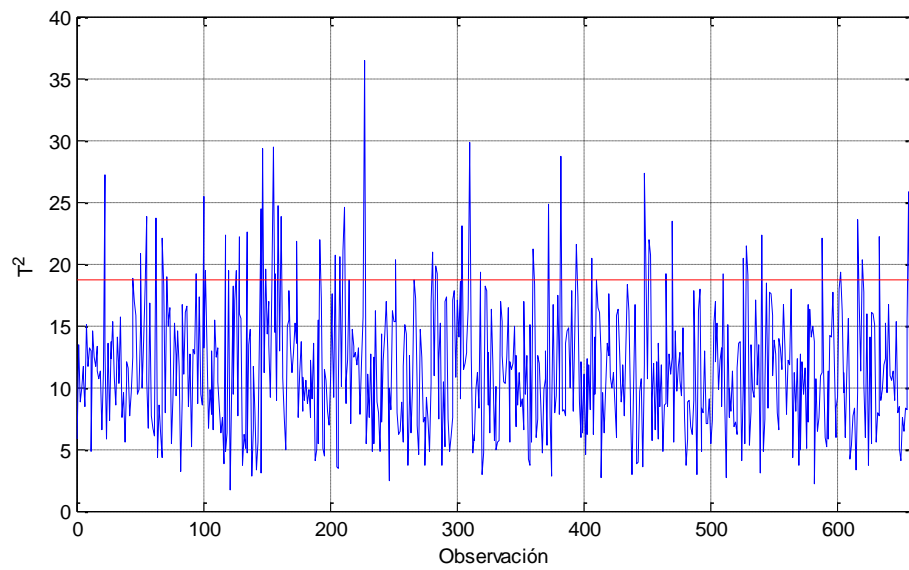
Los valores de  $k$  y  $r$  que minimizan la tasa de falsas alarmas en la ecuación (4.4) son  $k = 10$  y  $r = 6$ , por lo tanto el punto de operación óptimo del algoritmo (para el estadístico  $T^2$ ) de MAF-PCA aplicado al Tennessee Eastman Process se encuentra al remplazar:

$$(k, r) = (10, 6)$$

Se muestra, en la *Figura 13*, la respuesta del estadístico  $T^2$  y del umbral en el punto óptimo de operación usando el algoritmo MAF-PCA, la cual se compara con la respuesta para PCA convencional, mostrada en la *Figura 14*.



**Figura 13 – Respuesta del estadístico  $T^2$  y del umbral en el punto óptimo de operación con MAF-PCA.**



**Figura 14 - Respuesta del estadístico  $T^2$  con PCA convencional.**

La tasa de falsas alarmas para el algoritmo de MAF-PCA es de 7.36% mientras que para el algoritmo de PCA convencional es de 8.36%.

## 4.2. Estrategia Adaptativa con Umbral Dinámico para Estadístico $Q$

Se diseña ahora una estrategia adaptativa para el estadístico  $Q$  que es similar a la estrategia previamente diseñada para el estadístico  $T^2$ . Sea  $H_Q \in \mathbb{R}^{\hat{k}}$  la ventana móvil que contiene los últimos  $\hat{k}$  valores del estadístico  $Q$ ,

$$H_Q = \begin{bmatrix} Q_{i-\hat{k}+1} \\ \vdots \\ Q_{i-1} \\ Q_i \end{bmatrix} \quad (4.6)$$

La media,  $\bar{H}_Q$ , y la desviación estándar,  $\sigma(H_Q)$ , de la ventana móvil deben ser calculadas. Ahora, considérense los  $\hat{r}$  últimos valores del estadístico  $Q$ :  $Q_i, Q_{i-1}, \dots, Q_{i-\hat{r}+1}$ , con  $\hat{r} < \hat{k}$ . Si hay una todos los últimos  $\hat{r}$  valores consecutivos son mayores que el estadístico de prueba  $Q_\alpha$  se dice que hay una *racha*. Es evidente además que:

$$\hat{k} \in \mathbb{Z}^+ \quad \wedge \quad \hat{r} \in \mathbb{Z}^+ \quad (4.7)$$

Las reglas de decisión para el ajuste del umbral, para el estadístico  $Q$  son análogas a las diseñadas para el estadístico  $T^2$  y se definen a continuación:

- Si  $Q_i > Q_\alpha \wedge Q_{i-1} > Q_\alpha \wedge \dots \wedge Q_{i-\hat{r}+1} > Q_\alpha$ , entonces el umbral será  $Q_\alpha$  y una alarma será desencadenada.
- Si  $Q_i < Q_\alpha \vee Q_{i-1} < Q_\alpha \vee \dots \vee Q_{i-\hat{r}+1} < Q_\alpha$ , entonces:
  - Si  $Q_\alpha \geq \bar{H}_Q + \sigma(H_Q)$ , entonces el umbral será  $Q_\alpha$ .
  - Si  $Q_\alpha < \bar{H}_Q + \sigma(H_Q)$ , entonces el umbral será  $\bar{H}_Q + \sigma(H_Q)$ .

Se hace una ilustración con diagrama de flujo del algoritmo diseñado, con  $\hat{r} = 3$ , en la *Figura 15*. Como puede observarse el diagrama de flujo del algoritmo MAF-PCA para el estadístico  $Q$  es análogo al diagrama de flujo del mismo algoritmo para el estadístico  $T^2$ , que se ha desarrollado previamente. De nuevo, el bloque HS hace referencia a un “*High Selector*”.

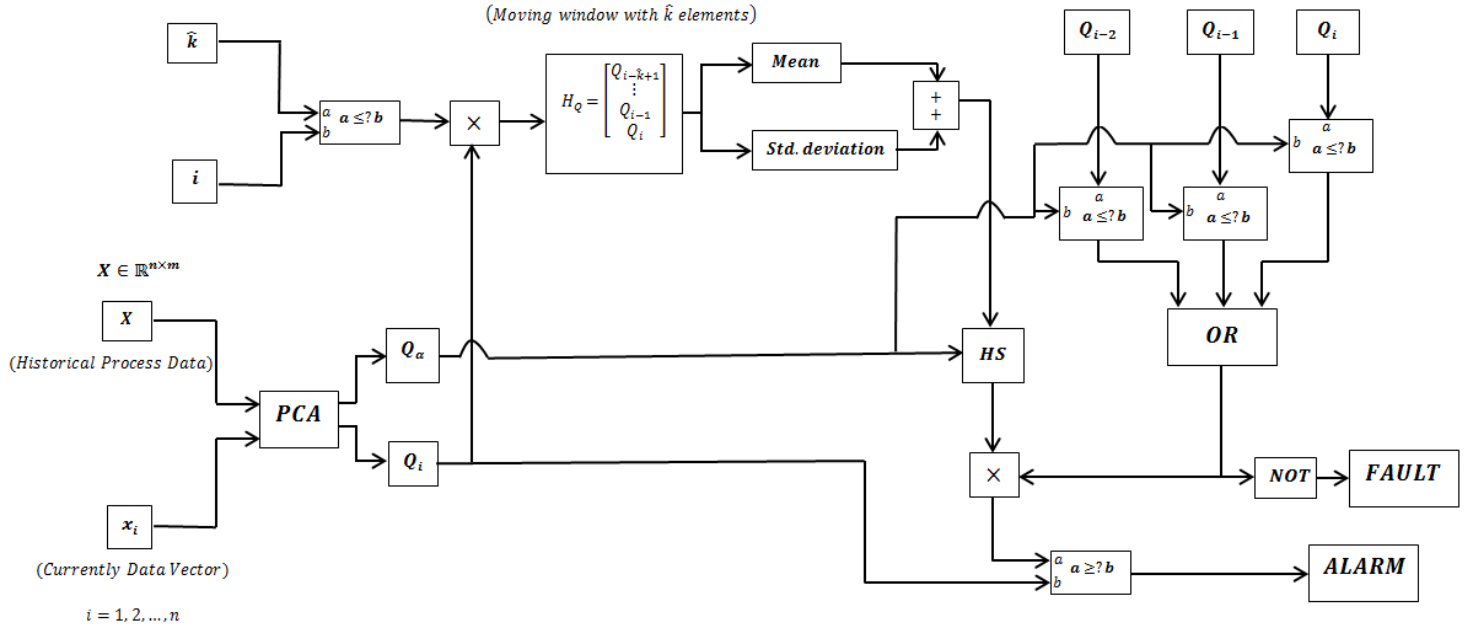


Figura 15 – Esquema del algoritmo MAF-PCA para el estadístico  $Q$ , con  $\hat{r} = 3$ .

Dado que se tienen, de igual manera, dos parámetros cuyo valor es variable,  $\hat{r}$  &  $\hat{k}$ , es necesario hacer otro diseño de experimento (DOE), con el fin de determinar los valores óptimos para estos parámetros. Se llevarán a cabo entonces dos experimentos factoriales con regiones de experimentación y niveles para  $\hat{r}$  &  $\hat{k}$  iguales a los establecidos para  $r$  &  $k$ , así como también un experimento de optimización.

#### 4.2.1. Experimento Factorial en la Región de Experimentación 1 (Para Estadístico $Q$ )

Al igual que para el estadístico  $T^2$ , se lleva a cabo el entrenamiento se utilizan datos del TEP con todas las variables de entrada en estado estable, si embargo, con el fin de inducir falsas alarmas, para la fase de detección se usan datos generados con perturbaciones de 0,05% respecto a la magnitud de estado estable de las variables manipuladas. Esto se hace para la región de experimentación 2 también. Se definen los niveles de los factores para la región de experimentación 1 en la *Tabla 12*.

Tabla 12 - Factores y niveles definidos para la región de experimentación 1. (Para estadístico  $Q$ ).

Experimento Factorial 1 (Para Estadístico $Q$ )			
Factor	Niveles		
	Bajo	Medio	Alto
Tamaño de Ventana ( $\hat{k}$ )	10	30	50
Tamaño de Racha ( $\hat{r}$ )	3	6	9

La variable de respuesta será la tasa de falsas alarmas, pero con respecto al estadístico  $Q$ . Dicha tasa de falsas alarmas se define como:

$$Tasa\ de\ F.A. = 100\% \times \left( \sum_{\forall i | Q_i > Q_\alpha} 1 \right) / \left( \sum_{\forall i} 1 \right) \quad (4.8)$$

El experimento de tres niveles con dos factores se lleva a cabo con una réplica ( $n = 2$ ). Se muestra en la *Tabla 13* el experimento aleatorizado, así como también el valor de la variable de respuesta para cada condición experimental.

Tabla 13 - Condiciones experimentales de la región de experimentación 1 (Para Estadístico  $Q$ ).

Réplica	( $\hat{k}$ )	( $\hat{r}$ )	Tasa de F.A. (%)
1	50	6	10.2564
1	30	3	8.7821
1	10	3	7.8205
1	10	6	7.9487
1	30	6	8.7821
1	30	9	8.5256
1	50	3	9.359
1	50	9	9.8718
1	10	9	6.859
2	30	3	9.2949
2	10	6	8.141
2	50	3	9.1026
2	50	6	9.7436
2	30	9	9.5513

2	50	9	9.6795
2	10	3	7.1154
2	30	6	8.4615
2	10	9	7.9441

Con ayuda del software Statgraphics® se lleva a cabo un análisis de varianza multifactorial (ANOVA multifactorial), cuyo resultado se muestra en la *Tabla 14*.

**Tabla 14 - ANOVA para la región de experimentación 1 (Para Estadístico  $Q$ ).**

Fuente	Suma de Cuadrados	Grados de Libertad	Cuadrado Medio	F	Valor-P ( $\alpha = 0.05$ )
$\hat{k}$	12.6135	2	6.30676	32.48	0.0001
$\hat{r}$	0.288012	2	0.144006	0.74	0.5033
$\hat{k} \cdot \hat{r}$	0.07009	4	0.267523	1.38	0.3157
Residuales	1.74754	9	0.194171		
Total (Corregido)	15.7192	17			

Como se puede observar, sólo el factor  $\hat{k}$  es significativo en la región de experimentación 1.

#### 4.2.2. Experimento Factorial en la Región de Experimentación 2 (Para Estadístico $Q$ )

Se definen los niveles de los factores para la región de experimentación 1 en la *Tabla 15*.

**Tabla 15 - Factores y niveles definidos para la región de experimentación 2. (Para estadístico  $Q$ ).**

Experimento Factorial 2 (Para Estadístico $Q$ )			
Factor	Niveles		
	Bajo	Medio	Alto
Tamaño de Ventana ( $\hat{k}$ )	50	70	90
Tamaño de Racha ( $\hat{r}$ )	3	11	19

El experimento de tres niveles con tres factores se lleva a cabo con una réplica. Se muestra en la *Tabla 16* el experimento aleatorizado, así como también el valor de la variable de respuesta para cada condición experimental.

**Tabla 16 - Condiciones experimentales de la región de experimentación 2 (Para Estadístico  $Q$ ).**

<b>Réplica</b>	<b><math>\hat{k}</math></b>	<b><math>\hat{r}</math></b>	<b>Tasa de FA (%)</b>
1	90	19	11.7117
1	70	11	10.5105
1	70	3	12.1622
1	50	11	10.9610
1	50	3	10.8108
1	90	11	12.7628
1	50	19	11.1111
1	70	19	12.3123
1	90	3	13.8138
2	90	11	12.3126
2	50	19	10.6607
2	50	3	12.7628
2	70	11	11.2613
2	90	3	13.3634
2	70	3	12.2519
2	70	19	13.0631
2	50	11	9.4595
2	90	19	12.6126

Con ayuda del software Statgraphics® se lleva a cabo un análisis de varianza multifactorial (ANOVA multifactorial), cuyo resultado se muestra en la *Tabla 17*.



Tabla 17 - ANOVA para la región de experimentación 2 (Para Estadístico  $Q$ ).

Fuente	Suma de Cuadrados	Grados de Libertad	Cuadrado Medio	F	Valor-P ( $\alpha = 0.05$ )
$\hat{k}$	9.7567	2	4.87835	10.19	0.0049
$\hat{r}$	5.20483	2	2.60219	5.43	0.0283
$\hat{k} \cdot \hat{r}$	2.96679	4	0.741699	1.55	0.2685
Residuales	4.31014	9	0.478904		
Total (Corregido)	22.238	17			

Como se puede observar en la *Tabla 17*, esta vez los factores  $\hat{k}$  y  $\hat{r}$  son significativos en la región de experimentación 2. Se procede ahora a encontrar el punto de operación óptimo.

#### 4.2.3. Optimización (Para Estadístico $Q$ )

Debido a que el factor  $\hat{k}$  es significativo en ambas regiones de experimentación y además el factor  $\hat{r}$  es significativo en la región de experimentación 2, es deseable llevar a cabo una experimento en una región de experimentación que sea la unión de la 1 y la 2. Existe, sin embargo, una restricción:  $\hat{k} > \hat{r}$ . Debido a lo anterior, se realiza la búsqueda del punto óptimo de operación en las dos regiones (1 y 2) por separado. Una vez se hayan encontrado los óptimos para ambas regiones de experimentación se determina cuál ofrece un valor de la variable de respuesta (tasa de falsas alarmas) menor y se declara dicho punto de operación.

Dado que ya se han llevado a cabo experimentos con puntos centrales en ambas regiones de experimentación, se utilizarán los datos de dichos experimentos para desarrollar modelos de regresión y evaluar así los puntos de operación óptimos en cada región de experimentación. La regresión para la región de experimentación 1 se hace exclusivamente en función del parámetro  $\hat{k}$  ya que fue el único que resultó significativo en dicha región. Se muestran los modelos de regresión desarrollados en la *Tabla 18*.

Tabla 18 - Resumen de modelos de regresión para optimización de la tasa de falsas alarmas para estadístico Q.

Modelo	Ajuste ( $R^2$ ajustado)	Significancia del modelo (en ANOVA)	¿Todos los coeficientes significativos?	¿Estructura en los residuales?
$7.21248 + 0.0507675\hat{k}$	0.774	OK	Si	NO
$4.50467 + 0.0424167k + 0.0004121k^2$	0.776	OK	NO	NO

Dicha regresión se hace con los datos de la *Tabla 13*, y se muestra la ecuación de regresión a continuación:

$$F.A.R. = 7.21248 + 0.0507675\hat{k} \quad (4.9)$$

Se muestran las estadísticas de la regresión en la *Tabla 19*.

Tabla 19 - Estadísticas de la regresión en la Región de experimentación 1 para estadístico Q.

Estadísticas de la regresión	
Coeficiente de correlación múltiple	0.88713896
Coeficiente de determinación $R^2$	0.78701553
$R^2$ ajustado	0.773704
Error típico	0.45743429
Observaciones	18

El análisis de varianza (ANOVA) para esta regresión se muestra en la *Tabla 20*.

Tabla 20 - ANOVA de la regresión de la tasa de falsas alarmas para el estadístico Q en la región de operación 1.

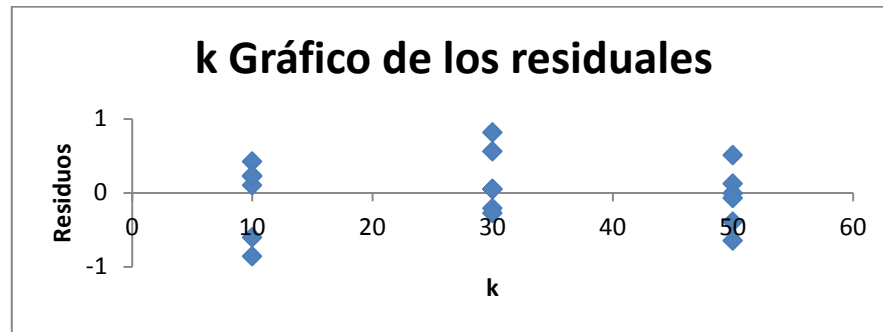
Fuente	Suma de cuadrados	Grados de Libertad	Cuadrado Medio	F	Valor-P ( $\alpha = 0.05$ )
Regresión	12.37122747	1	12.37122747	59.1228485	9.23773E-07
Residuos	3.347938139	16	0.209246134		
Total	15.71916561	17			

La evaluación estadística para los coeficientes de regresión se muestra en la *Tabla 21*.

**Tabla 21 - Análisis de los coeficientes de regresión en la Región de experimentación 1 para estadístico  $T^2$ .**

	<b><i>Coefficientes</i></b>	<b><i>Error típico</i></b>	<b><i>Estadístico t</i></b>	<b><i>Valor-P (<math>\alpha = 0.05</math>)</i></b>
Intercepción	7.212480556	0.22551815	31.9818188	6.2614E-16
$\hat{k}$	0.0507675	0.0066025	7.68913835	9.2377E-07

Se muestran los residuales en la *Figura 16*. Se observa que no posee una estructura evidente.



**Figura 16 - Gráfico de residuales para regresión en la región de experimentación 1 para estadístico  $Q$ .**

Debido a que con un modelo lineal se obtuvo un buen ajuste y se obtuvo significancia del modelo de regresión, así como también de cada uno de los coeficientes de regresión, y además no se tiene una estructura en los residuales, no se llevaron a cabo modelos de distinta naturaleza.

La regresión para la región de experimentación 2 se debe hacer con los datos de la *Tabla 16*. Se muestra en la *Tabla 22* el resumen de los modelos de regresión realizados.

**Tabla 22 - Modelos de regresión para la tasa de falsas alarmas en la Región de experimentación 2 para estadístico Q.**

<b>Modelo</b>	<b><math>R^2</math> ajustado</b>	<b>¿Modelo significativo?</b>	<b>¿Todos los coeficientes significativos?</b>
$9.15355 + 0.045046\hat{k} - 0.038473\hat{r}$	0.420	OK	NO
$8.895617 + 0.045046\hat{k} - 0.001009\hat{r}^2$	0.383	OK	NO
$8.40083 + 0.06779\hat{k} - 0.00016246\hat{k}^2 - 0.384729\hat{r}$	0.380	OK	NO
$10.3878148 + 0.04504583\hat{k} - 0.38511615\hat{r} + 0.01575651\hat{r}^2$	0.601	OK	OK

Se tiene entonces que la mejor regresión encontrada es:

$$10.3878148 + 0.04504583\hat{k} - 0.38511615\hat{r} + 0.01575651\hat{r}^2 \quad (4.10)$$

Se muestran las estadísticas de la regresión en la *Tabla 23*.

**Tabla 23 - Estadísticas de la regresión en la Región de experimentación 2 para estadístico Q.**

<b>Estadísticas de la regresión</b>	
Coeficiente de correlación múltiple	0.81976282
Coeficiente de determinación $R^2$	0.67201109
$R^2$ ajustado	0.60172775
Error típico	0.72179454
Observaciones	18

La El análisis de varianza (ANOVA) para esta regresión se muestra en la *Tabla 24*.

Tabla 24 - ANOVA de la regresión de la tasa de falsas alarmas para el estadístico Q en la región de operación 1.

<i>Fuente</i>	<i>Suma de cuadrados</i>	<i>Grados de Libertad</i>	<i>Cuadrado Medio</i>	<i>F</i>	<i>Valor-P</i> ( $\alpha = 0.05$ )
Regresión	14,94419374	3	4,981397914	9,56145648	0,001083042
Residuos	7,293822962	14	0,520987354		
Total	22,2380167	17			

La evaluación estadística para los coeficientes de regresión se muestra en la *Tabla 25*.

Tabla 25 - Análisis de los coeficientes de regresión en la Región de experimentación 2 para estadístico  $T^2$ .

	<i>Coefficientes</i>	<i>Error típico</i>	<i>Estadístico t</i>	<i>Valor-P</i>
Intercepción	10,3878148	0,91541632	11,34764	$1,9102 \times 10^{-8}$
$\hat{k}$	0,04504583	0,01041821	4,32376074	0,00070061
$\hat{\rho}^2$	0,01575651	0,00563902	2,79419312	0,0143433
$\hat{\rho}$	-0,38511615	0,12676303	-3,03807948	0,00885682

Se muestran los residuales para  $\hat{k}$  y  $\hat{\rho}$  en la *Figura 17* y la *Figura 18* respectivamente. Se observa que no poseen una estructura evidente en ninguno de los dos casos.

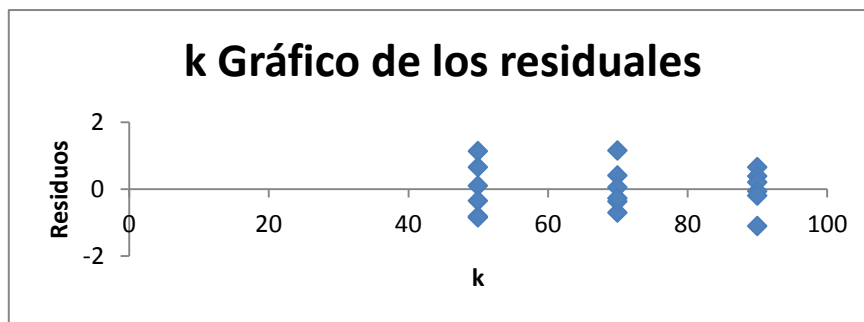


Figura 17 - Gráfico de residuales de  $\hat{k}$  para regresión en la región de experimentación 2 para estadístico Q.

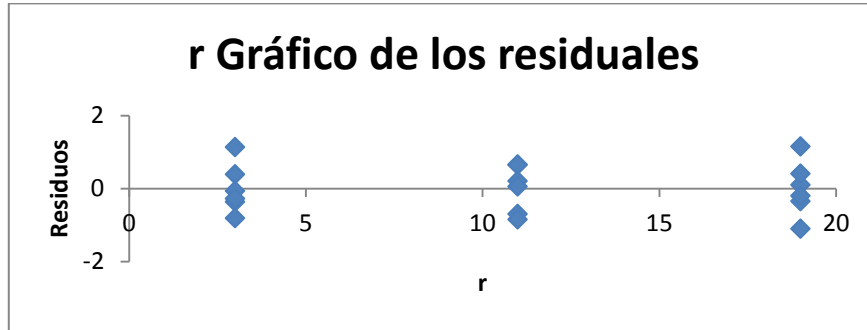


Figura 18 - Gráfico de residuales de  $\hat{r}$  para regresión en la región de experimentación 2 para estadístico  $Q$ .

Para encontrar el punto de operación óptimo en la región de experimentación 1 se busca el mínimo de la ecuación (4.9), la cual es la ecuación de regresión para dicha región de experimentación. La optimización se hace con la restricción:

$$10 \leq \hat{k} \leq 50 \quad (4.11)$$

El valor de  $\hat{k}$  que minimiza la ecuación (4.9) es  $\hat{k} = 10$ , y la tasa de falsas alarmas correspondiente es  $F.A.R. = 7.72\%$ . Se puede fijar  $\hat{r} = 6$ , el cual es el valor medio del rango definido para dicho parámetro, ya que no es significativo en esta región de experimentación.

Se debe encontrar ahora el punto óptimo de operación en la región de experimentación 2, para lo que se busca minimizar la ecuación (4.10), la cual es la ecuación de regresión para dicha región de experimentación. La optimización se hace con la restricción:

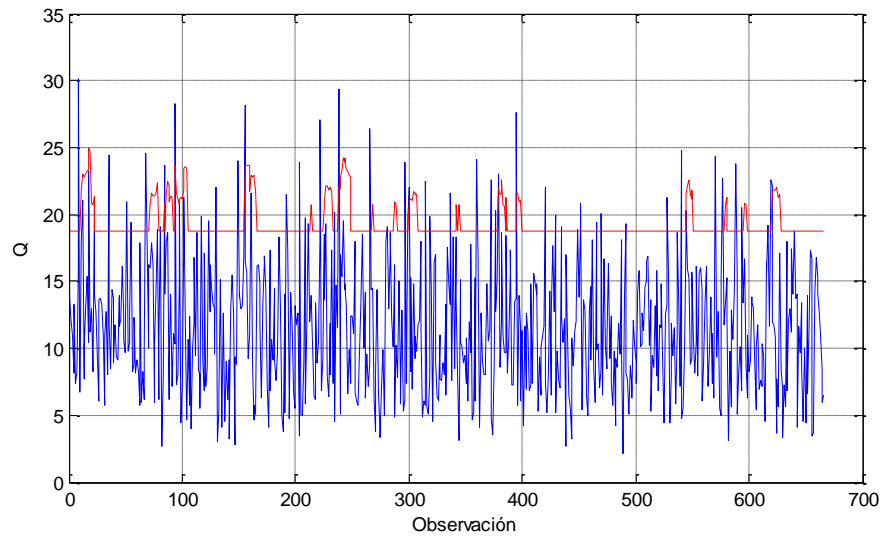
$$50 \leq \hat{k} \leq 90, \quad 3 \leq \hat{r} \leq 19 \quad (4.12)$$

Los valores de  $\hat{k}$  y  $\hat{r}$  que minimiza la ecuación (4.10) son  $\hat{k} = 50$  y  $\hat{r} = 11$ , y la tasa de falsas alarmas correspondiente es  $F.A.R. = 10.31\%$ .

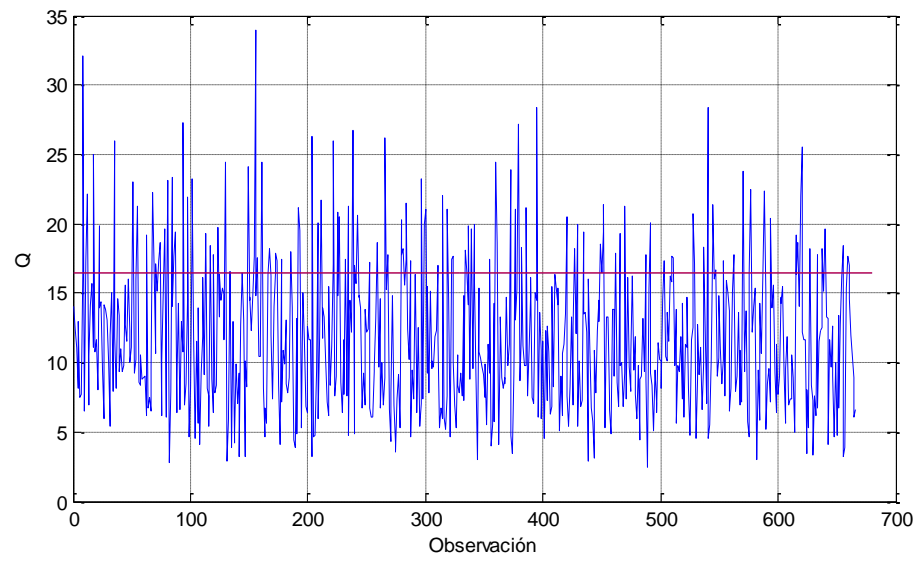
Si se compara el valor de la tasa de falsas alarmas obtenida en el punto de operación óptimo de la región de experimentación 1 con el valor correspondiente al punto de operación óptimo de la región de experimentación 2, es evidente que la tasa de falsas es menor en el punto óptimo de operación de la región de experimentación 1. Debido a lo anterior se concluye que el punto óptimo de operación será:

$$(\hat{k}, \hat{r}) = (10, 6)$$

Se muestra, en la *Figura 19*, la respuesta del estadístico  $Q$  y del umbral en el punto óptimo de operación usando el algoritmo MAF-PCA, la cual se compara con la respuesta para PCA convencional, mostrada en la *Figura 20*. La tasa de falsas alarmas para el algoritmo de MAF-PCA es de 10.21% mientras que para el algoritmo de PCA convencional es de 13.83%.



**Figura 19 – Respuesta del estadístico Q y el umbral con el enfoque MAF-PCA.**



**Figura 20 – Respuesta del estadístico Q con PCA convencional.**



## 5. VALIDACIÓN DE TÉCNICAS DESARROLLADAS EN UN PROCESO DE REFERENCIA

Se han desarrollado en esta investigación dos técnicas adaptativas de detección de fallas: PCA recursivo adaptativo ponderado (WARP) y PCA adaptativo con umbral dinámico (MAF-PCA). La validación de las dos técnicas desarrolladas es llevada a cabo sobre un proceso de referencia que ha sido ampliamente utilizado para la evaluación de técnicas de detección y diagnóstico de fallas: el Tennessee Eastman Process (TEP) [37]. Se comparará el desempeño de los enfoques desarrollados con el desempeño del algoritmo convencional de PC con el fin de cuantificar las mejoras conseguidas mediante la medición de la tasa de falsas alarmas para los estadísticos  $T^2$  y  $Q$  respectivamente, tal como se ha definido en el capítulo anterior en las ecuaciones (4.3) y (4.8).

Este capítulo está dividido en tres secciones: en la primera sección se describe el proceso de referencia (TEP), en la segunda sección se lleva a cabo la validación de la técnica de PCA adaptativo recursivo ponderado y finalmente en la tercera sección se lleva a cabo la validación de la técnica de PCA adaptativo con umbral dinámico.

### 5.1. Proceso de Referencia: Tennessee Eastman Process (TEP)

Como se mencionó anteriormente, se usará como proceso de referencia o “*Benchmark Process*” el Tennessee Eastman Process (TEP). El TEP fue creado por *Eastman Chemical Company* con el fin de proveer un proceso industrial realista para la evaluación de métodos de control y monitoreo de procesos [37]. Este proceso está basado en la simulación de un proceso químico real, en el cual los componentes, cinéticas y las condiciones de operación han sido modificados por razones de propiedad intelectual.

#### 5.1.1. Descripción del Proceso

Con el fin de ilustrar el proceso, un P&ID que contiene los componentes del Tennessee Eastman Process se muestra en la *Figura 21*.

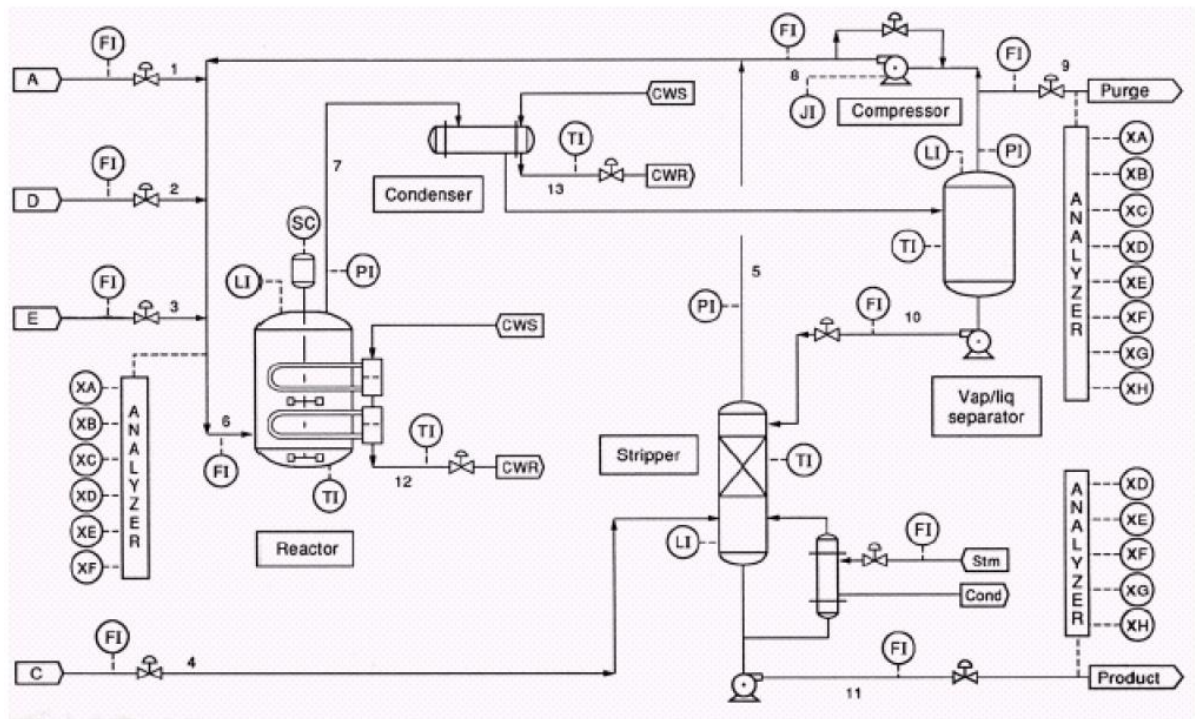
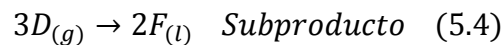
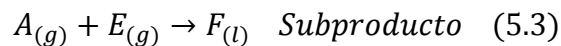
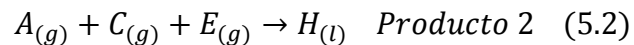
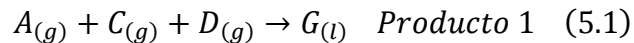


Figura 21 – Diagrama de flujo del Tennessee Eastman Process (TEP).

El proceso tiene cuatro unidades operacionales: un reactor exotérmico de dos fases, un separador flash, un extractor con re-hervidor y un compresor de recirculación. El proceso es alimentado por los reactantes gaseosos *A*, *C*, *D* y *E* y el inerte *B*, mientras que se forman los productos líquidos *G* y *H*. Las reacciones que se llevan a cabo en el reactor se muestran a continuación:



La especie *F* es un subproducto de las reacciones. Las reacciones son irreversibles, exotérmicas y aproximadamente de primer orden con respecto a las concentraciones de los reactantes. Las tasas de reacción son ecuaciones de Arrhenius en función de la

temperatura, donde la reacción para  $G$  tiene una energía de activación mayor que la reacción para  $H$ . La corriente de producto del reactor es enfriada mediante un condensador y luego alimentada hacia un separador líquido-vapor. El vapor que sale del separador es recirculado a la alimentación del reactor mediante un compresor. Una parte la corriente de reciclo es purgada para evitar la acumulación del inerte y el subproducto en el proceso. Los componentes condensados del separador (corriente 10) es bombeada a un separador. La corriente 4 es usada para separar los reactantes remanentes de la corrientes 10, los cuales son combinados con la corriente de reciclo a través de la corriente 5. Los productos  $G$  y  $H$ , que salen de la base del separador, se envían a un proceso aguas abajo que no está incluido en el diagrama.

Las ecuaciones que modelan el Tennessee Eastman Process se pueden dividir en ecuaciones de estado y en ecuaciones de salida. Este modelo es descrito detalladamente en los anexos, incluyendo todas las ecuaciones y la nomenclatura correspondiente.

## 5.2. Validación de Estrategia WRPDA en Proceso de Referencia

Debido a la cantidad de operaciones matriciales que requiere la estrategia diseñada, es deseable que las matrices con las que se va a trabajar estén bien condicionadas, es decir, que no presenten dependencias lineales entre sus filas o columnas (en la fase de entrenamiento) y que sus variables tengan varianzas significativamente mayores que cero. Debido a que las matrices de covarianza  $S_i$  provienen de la operación  $S_i = \frac{1}{n_i-1} X_i^T X_i$  serán siempre simétricas, lo que implican que son diagonalizables y su determinante satisface:

$$\det(S_i) \neq 0, \quad \forall i \quad (5.1)$$

No obstante, si el determinante se acerca mucho a cero, esto inducirá errores. Debido a lo anterior se calcula la matriz de correlación de Pearson para los datos contenidos en el training set inicial  $S_0$ . Dicha matriz  $C \in \mathbb{R}^{41 \times 41}$  es muy grande, por lo que se mostrará en la *Tabla 26* una lista de aquellos pares de variables cuyo coeficiente de correlación supere, en valor absoluto, el 95% (0.95), omitiendo la correlación de cada variable con ella misma ya que siempre será 1 y se encuentran en la diagonal de la matriz.

**Tabla 26 – Pares de variables cuyo coeficiente de correlación**

<b>Variable 1</b>	<b>Variable 2</b>
XMEAS(37)	XMEAS(29)
XMEAS(37)	XMEAS(41)
XMEAS(38)	XMEAS(39)
XMEAS(39)	XMEAS(41)

De acuerdo a los resultados de la Tabla 26 se puede prescindir de las variables XMEAS(38), XEMAS(39) y XMEAS(41) para el entrenamiento y la detección de fallas ya que aportan información redundante. Se debe prescindir también de aquellas variables cuya varianza sea cercana a cero. Se listan a continuación aquellas variables cuya varianza natural sea menor que 0.0001: XMEAS(1), XMEAS(9), XMEAS(10) y XMEAS(22). Se muestran en la *Tabla 27* todas las variables que serán excluidas.

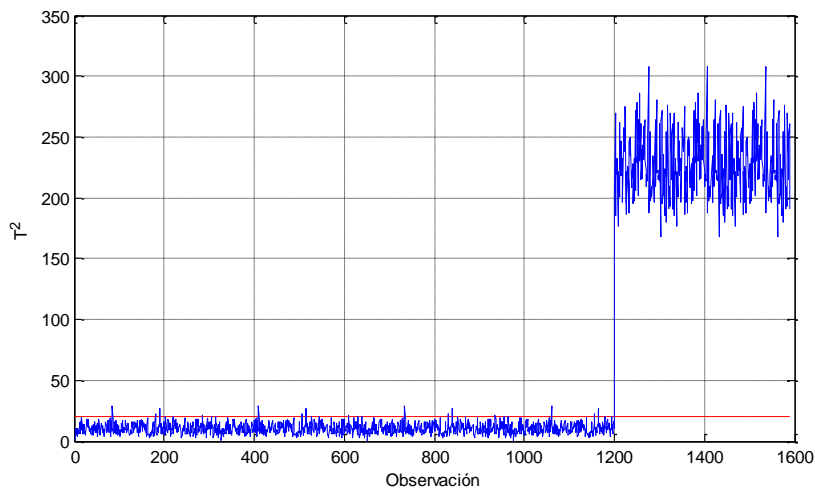
**Tabla 27 – Variables excluidas.**

<b>Variables Excluidas</b>	
1	XMEAS(1)
2	XMEAS(9)
3	XMEAS(10)
4	XMEAS(22)
5	XMEAS(29)
6	XMEAS(38)
7	XMEAS(39)
8	XEMAS(41)

Para llevar a cabo la validación de la estrategia de WRPCA se lleva a cabo un entrenamiento con datos del TEP con sus variables de entrada (XMV(1), XMV(2), ..., XMV(12)) en estado estable. Se inicia la detección de fallas con el proceso

inicialmente con las variables de entrada en estado estable. Se hace necesario inducir perturbaciones que provoquen cambios sutiles en el comportamiento del proceso, los cuales disparen falsas alarmas, después de lo cual se lleva a cabo la actualización recursiva y se mide la tasa de falsas alarmas. Se hacen pasos escalón de +0.25% en las variables  $XMV(2)$ ,  $XMV(6)$  y  $XMV(12)$  en el instante de muestreo  $t = 1200$ .

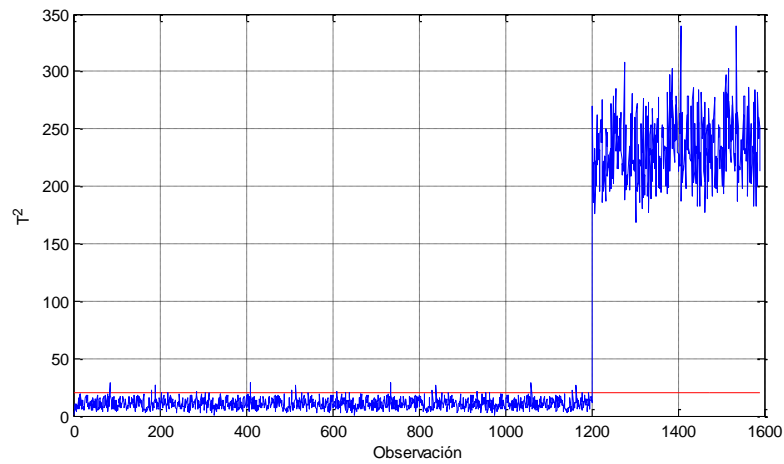
Se implementa en primer lugar el algoritmo de PCA convencional y se somete el proceso a monitoreo a través del mismo. Se mide la tasa de falsas alarmas del estadístico  $T^2$  a partir del instante de muestro 1200. Se ilustra el desempeño de PCA convencional en la *Figura 22*.



**Figura 22 – Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo WARP).**

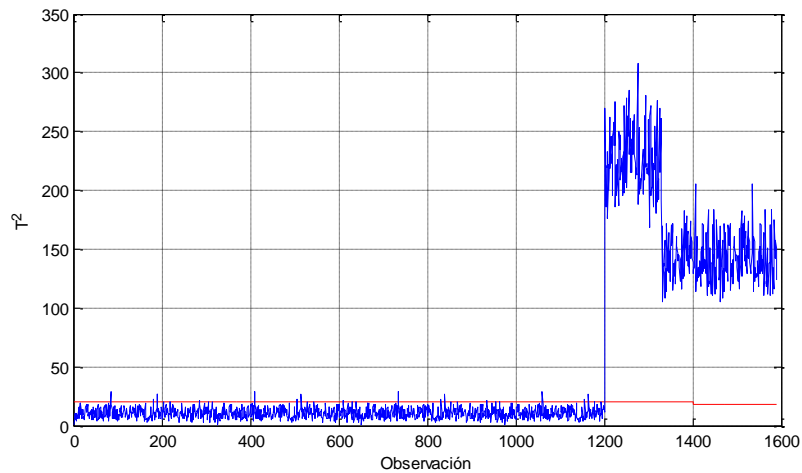
La tasa de falsas alarmas del estadístico  $T^2$  luego del instante de muestro 1200 es del 100%. Se implementa ahora el algoritmo de PCA adaptativo recursivo ponderado (WARP), variándose el factor de aprendizaje  $w$  desde 0 hasta 1 tomando los siguientes valores:  $w = 0, 0.1, \dots, 0.9, 1$ .

En todos los casos la actualización recursiva será llevada a cabo en el instante de muestreo  $t = 1350$ . El desempeño del algoritmo WARP con  $w = 0$  se ilustra en la *Figura 23*. La tasa de falsas alarmas después del instante de muestreo 1200 es del 100%, al igual que con PCA convencional.



**Figura 23 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0$ .**

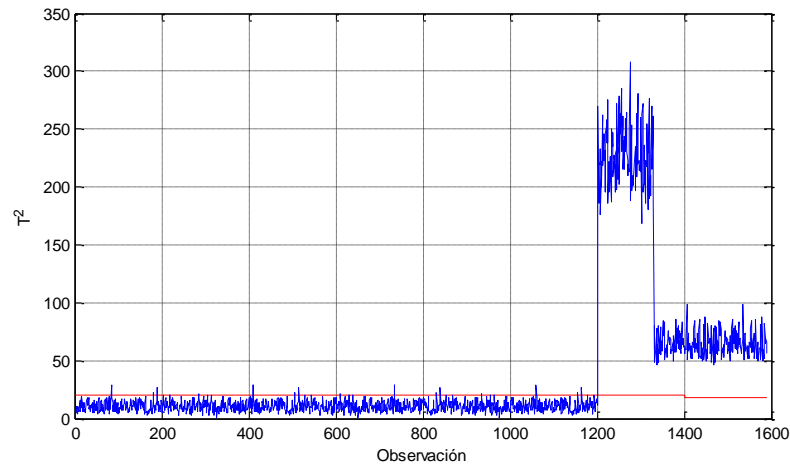
Además de que la tasa de falsas alarmas al aplicar el algoritmo WARP con  $w = 0$  es del 100% al igual que al aplicar el algoritmo de PCA convencional, el esquema de monitoreo o *Run Chart* es cualitativamente idéntico, ya que el factor de aprendizaje es 0 y la actualización recursiva no tiene efecto alguno. Se muestra el desempeño del algoritmo WARP con  $w = 0.1$  en la *Figura 24*.



**Figura 24 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.1$ .**

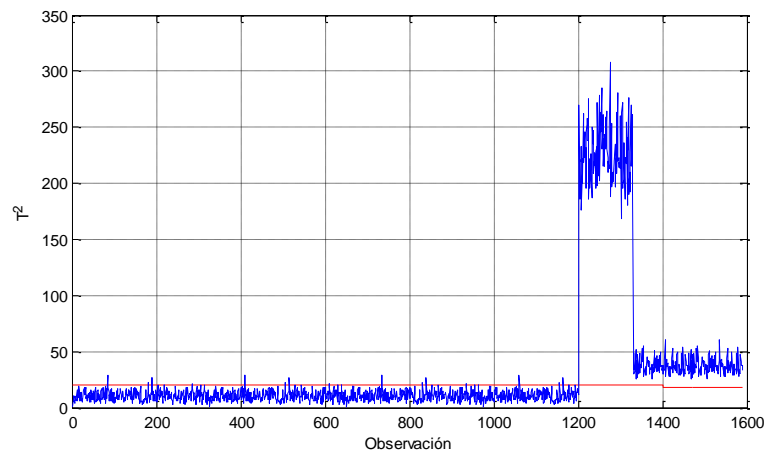
La tasa de falsas alarmas luego del instante de muestreo **1200** con  $w = 0.1$  sigue siendo del **100%**, sin embargo es notable que luego de la actualización recursiva, en

el instante de muestreo 1350, hay un cambio en el comportamiento del estadístico  $T^2$  y del umbral. Se ilustra en la *Figura 25* el desempeño para  $w = 0.2$ .



**Figura 25 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.2$ .**

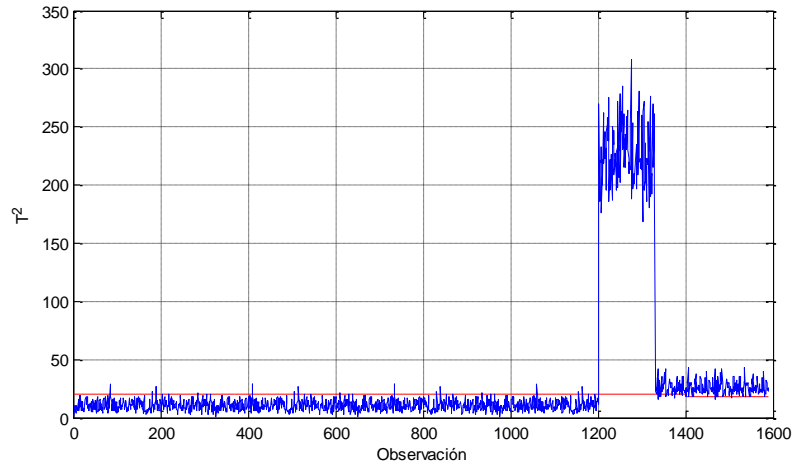
La tasa de falsas alarmas luego de la actualización recursiva sigue siendo del 100%, sin embargo el cambio en el estadístico  $T^2$  es más notable. Se ilustra en la *Figura 26* el desempeño para  $w = 0.3$ .



**Figura 26 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.3$ .**

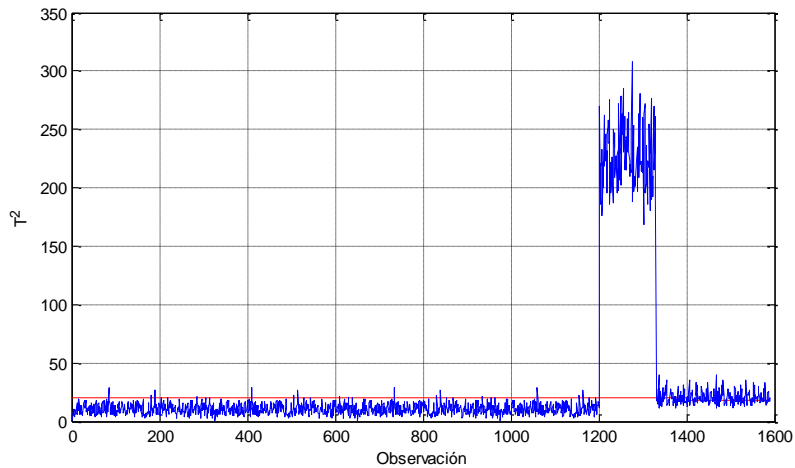
Se puede apreciar en la *Figura 26* que a pesar de la actualización recursiva con  $w = 0.3$  este factor de aprendizaje aún no es lo suficientemente grande para conllevar

a una reducción de la tasa de falsas alarmas. Se muestra el desempeño para  $w = 0.4$  en la *Figura 27*.



**Figura 27 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.4$ .**

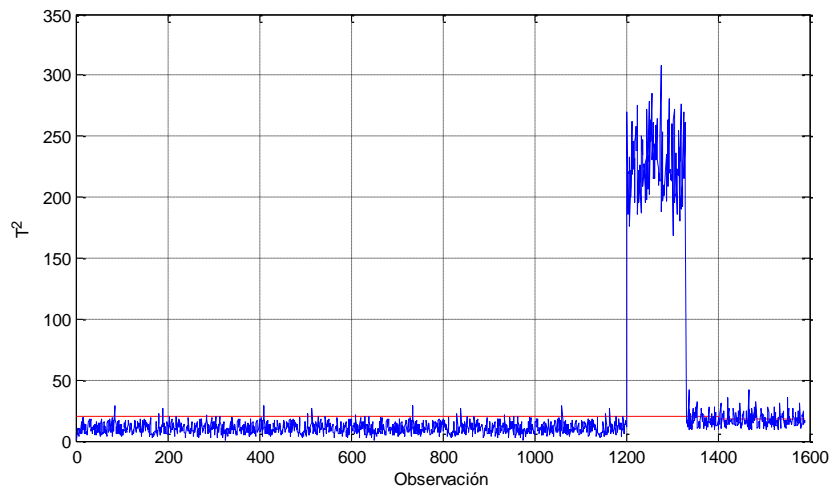
La tasa de falsas alarmas con  $w = 0.4$  es de 91.3043%. Se necesita un factor de aprendizaje mayor para que la reducción de la tasa de falsas alarmas sea sustancial. Se muestra en la *Figura 28* el desempeño para  $w = 0.5$ .



**Figura 28 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.5$ .**

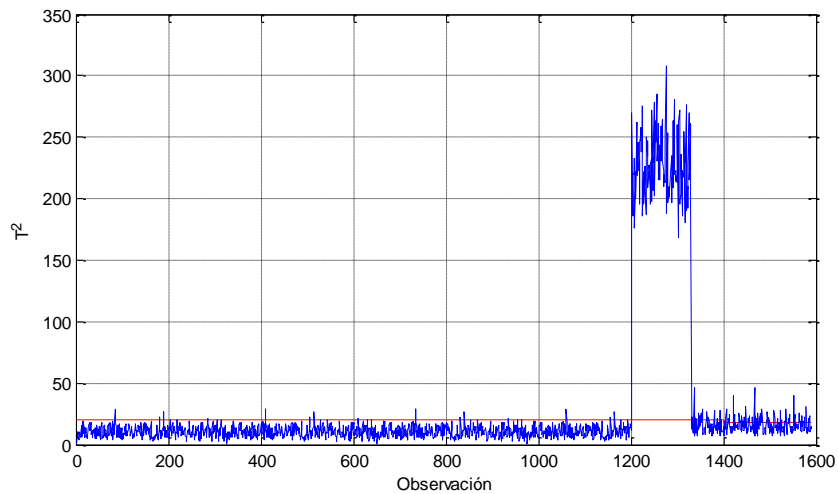
Con  $w = 0.5$  la tasa de falsas alarmas luego de la actualización recursiva es de 58.3851%. Se ilustra en la *Figura 29* el desempeño para  $w = 0.6$ .





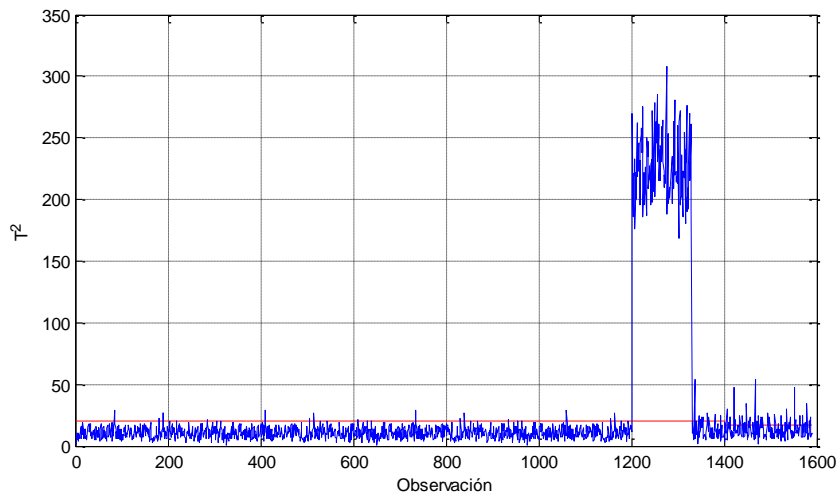
**Figura 29 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.6$ .**

La tasa de falsas alarmas para  $w = 0.6$  es de 37.2671%. Se muestra en la *Figura 30* el desempeño para  $w = 0.7$ .



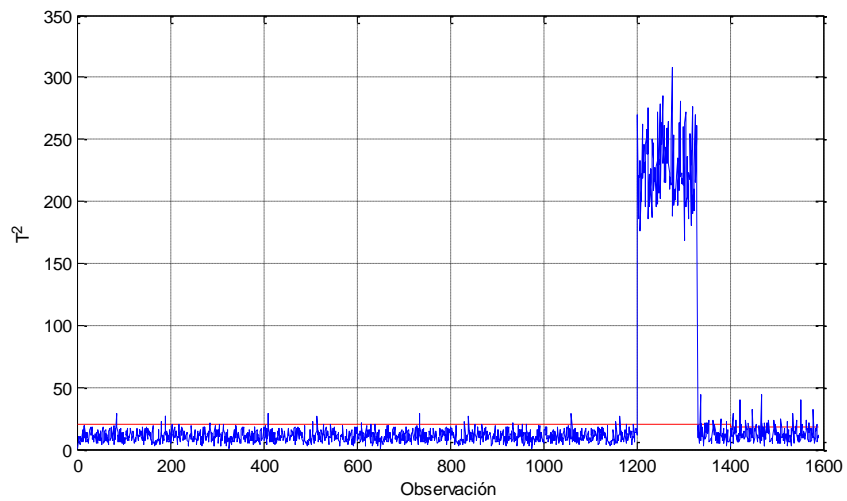
**Figura 30 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.7$ .**

La tasa de falsas alarmas para  $w = 0.7$  es de 28.5714%. El desempeño para  $w = 0.9$  se muestra en la *Figura 31*.



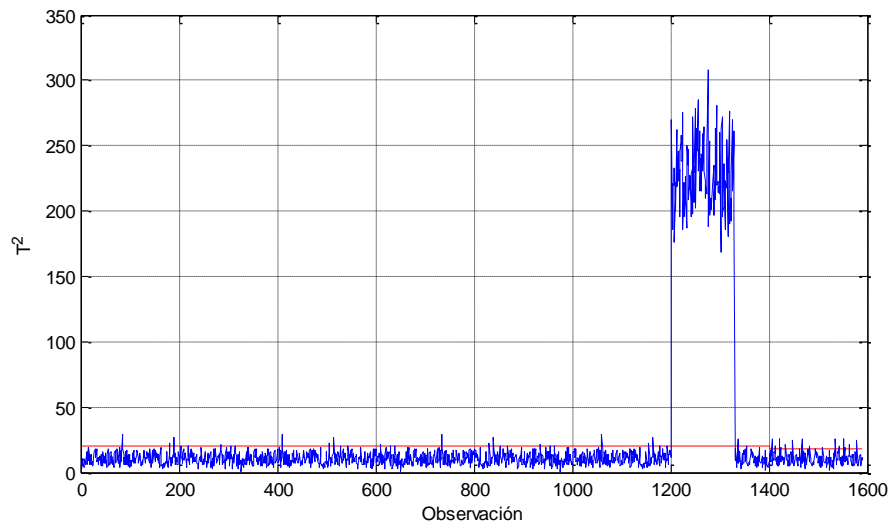
**Figura 31 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.9$ .**

La tasa de falsas alarmas para  $w=0.9$  es de 27.2741%. El desempeño para  $w = 0.95$  se muestra en la *Figura 32*.



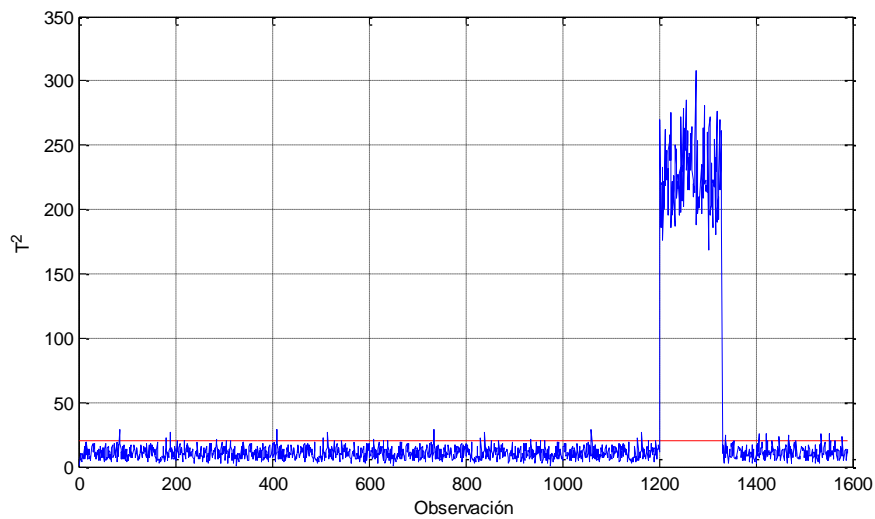
**Figura 32 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.95$ .**

La tasa de falsas alarmas para  $w = 0.95$  es del 14.2857%. El desempeño para  $w = 0.99$  se muestra en la *Figura 33*.



**Figura 33 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=0.99$ .**

La tasa de falsas alarmas para  $w = 0.99$  es del 8.5957%. Se evalúa finalmente el desempeño para  $w = 1$ , el cual se muestra en la *Figura 34*.



**Figura 34 - Desempeño del algoritmo WARP ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 con  $w=1$ .**

Al utilizar un factor de aprendizaje de  $w = 1$  se olvida la información inicial con la que se llevó a cabo el entrenamiento inicial y dicha información es remplazada por los nuevos datos que habían ocasionado el incremento en la tasa de falsas alarmas. La

tasa de falsas alarmas con resultante es del 5.8323%. Se resume el comportamiento de la tasa de falsas alarmas en función del factor de aprendizaje  $w$  en la *Figura 35*.

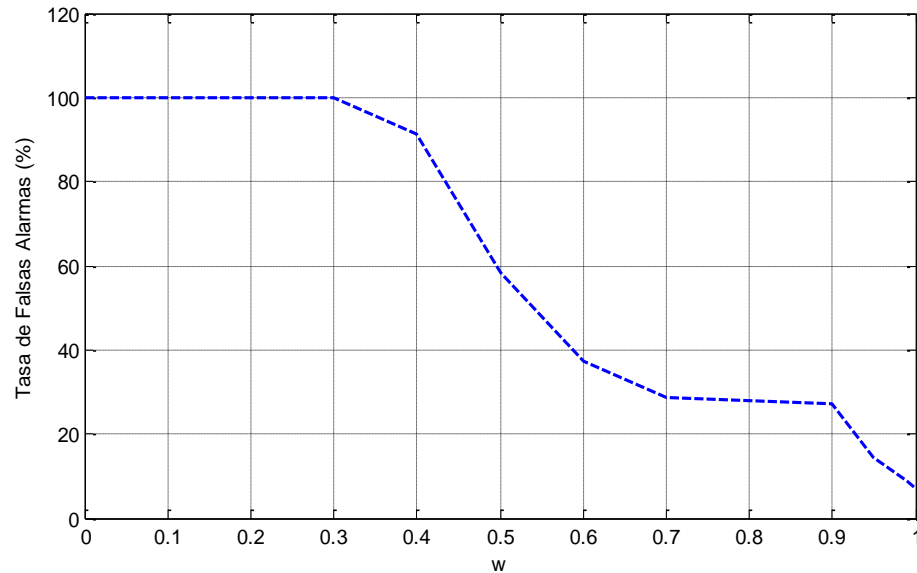
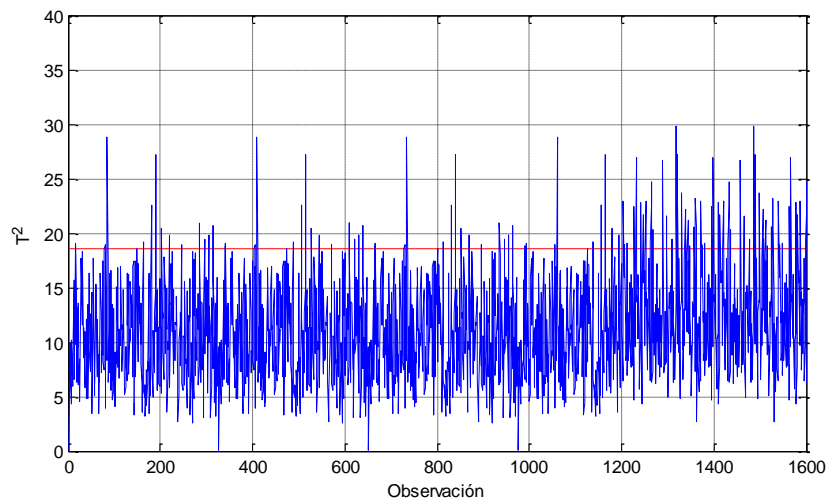


Figura 35 – Desempeño del algoritmo WARP en función del factor de aprendizaje  $w$ .

### 5.3. Validación de Estrategia Adaptativa con Umbral Dinámico; Moving Average Filter PCA (MAF-PCA), en Proceso de Referencia

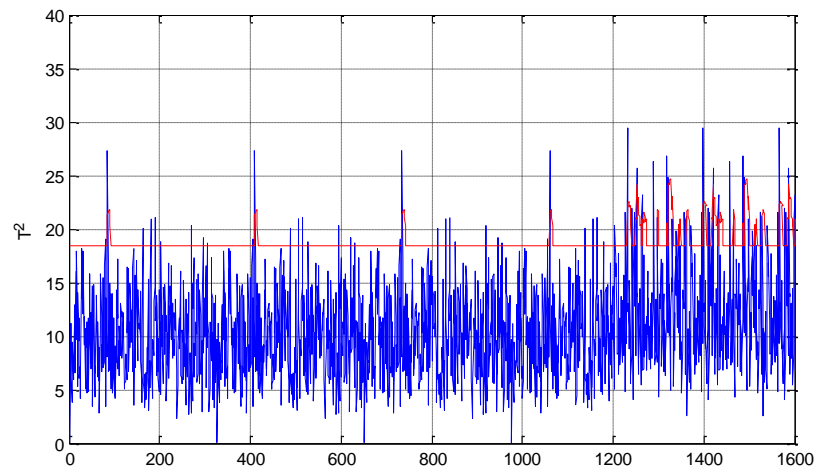
El desempeño del algoritmo de MAF-PCA se evalúa también en el TEP. En este caso también se hace necesario inducir perturbaciones que provoquen cambios en el comportamiento del proceso con el fin de poner a prueba el algoritmo. Los cambios deben ser más sutiles que los inducidos para evaluar el algoritmo WARP ya que el algoritmo MAF-PCA no realiza cambios en la estructura estadística del modelo de detección sino que modificará exclusivamente el umbral de detección. Se hacen entonces pasos escalón de  $+0.05\%$  en las variables  $XMV(2)$ ,  $XMV(6)$  y  $XMV(12)$  en el instante de muestreo  $t = 1200$ .

Se muestra en primer lugar, en la *Figura 36*, el desempeño del algoritmo de PCA convencional.



**Figura 36 - Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo MAF-PCA).**

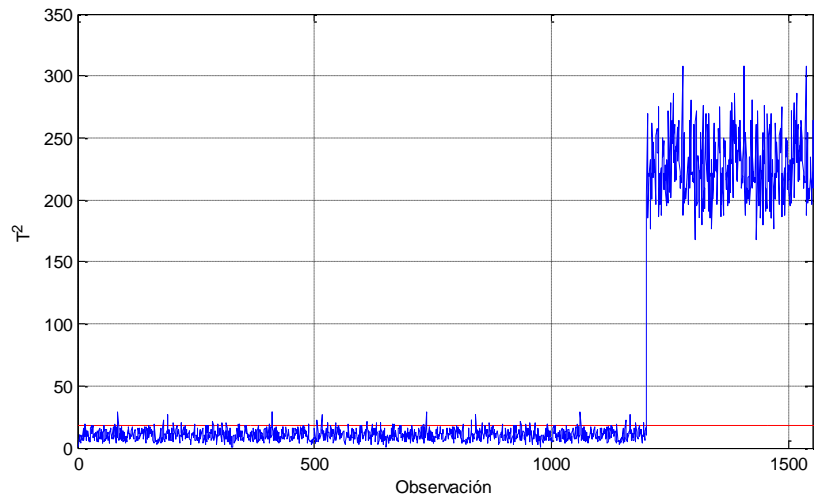
La tasa global de falsas alarmas es del 7.9812%. Se utiliza a continuación el algoritmo MAF-PCA, cuyo desempeño se muestra en la *Figura 37*.



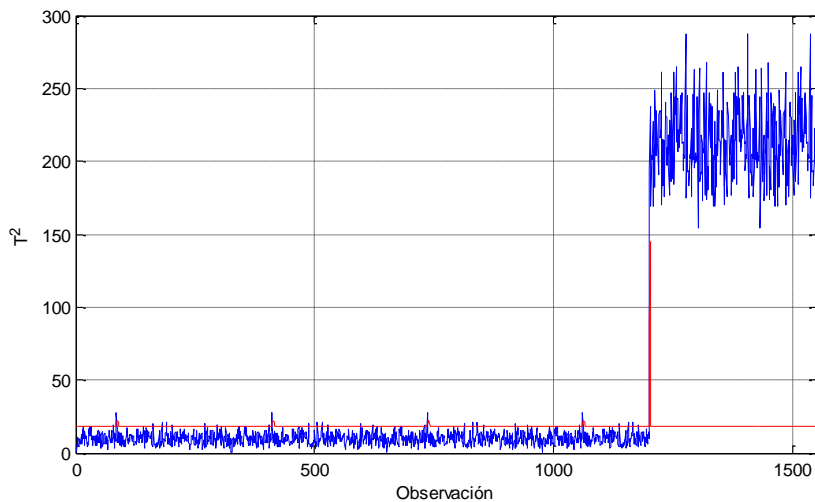
**Figura 37 - Desempeño del algoritmo MAF-PCA ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200.**

La tasa global de falsas alarmas del algoritmo MAF-PCA es del 4.7534%. Hay una reducción sustancial mediante la implementación de la estrategia diseñada. Se

someterá ahora el algoritmo ante un cambio en la estructura estadística del proceso mediante pasos escalón de  $+0.25\%$  en las variables  $XMV(2)$ ,  $XMV(6)$  y  $XMV(12)$  en el instante de muestreo  $t = 1200$ . Se muestra el desempeño del algoritmo de PCA convencional y el algoritmo de MAF-PCA en la *Figura 38* y la *Figura 39* respectivamente.



**Figura 38 - Desempeño de PCA convencional ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200 (para validación de algoritmo MAF-PCA).**



**Figura 39 - Desempeño del algoritmo MAF-PCA ante cambio de comportamiento inducido en el proceso de validación (TEP) en el instante de muestreo 1200.**

Bajo estas condiciones (un cambio sustancial en la estructura estadística subyacente del proceso) la tasa de falsas alarmas del algoritmo de PCA fue del 27.9873% frente a un 26.6038% para el algoritmo de MAF-PCA. El algoritmo de MAF-PCA fue “desactivado” mediante el filtro (reglas de decisión) una vez que una racha de violaciones del umbral ocurrió, por lo que no incluyó los nuevos datos luego del cambio en  $t = 1200$  en la ventana móvil y dejó invariante el valor del estadístico en  $T_{\alpha}^2$ , tal como es calculado por el algoritmo de PCA convencional.

## 6. CONCLUSIONES

A partir de la revisión del estado del arte se observó que una desventaja notable de los métodos convencionales de detección de fallas, incluyendo a PCA, es que implícitamente asumen que las observaciones son estadísticamente independientes de otras observaciones hechas en instantes de tiempo distintos, i.e., que los procesos no sufren derivas temporales o cambios en su estructura estadística subyacente con el paso del tiempo, por lo que una vez estructurados los modelos de detección de fallas en la fase de entrenamiento (con los datos históricos de proceso) éstos permanecen invariantes, mientras que la mayoría de los procesos reales muestran de hecho cambios de comportamiento en el tiempo. Cuando PCA es aplicado con modelos de detección de fallas fijos para el monitoreo de procesos con cambios o derivas temporales normales se tendrán eventualmente altas tasas de falsas alarmas.

Con el fin de abordar la problemática anteriormente expuesta se desarrolló, en primer lugar, una técnica de detección de fallas basada en PCA que permite la inclusión de nueva información disponible de operación de procesos mediante ecuaciones recursivas, las cuales permiten una re-estructuración de modelos de detección de fallas sin que se haga necesario hacer un re-entrenamiento (de manera convencional) que incluya tanto los datos históricos como los nuevos datos que se deseen incorporar. Luego de desarrollarse las fórmulas recursivas se introdujo un factor de ponderación o factor de aprendizaje  $w$ , el cual permite que se le asigne un peso a los nuevos datos cuya información se desea incorporar. Lo anterior permite modular el “aprendizaje” que el algoritmo realiza de los datos nuevos (respecto a los datos históricos), brindándole al usuario un parámetro al cual puede asignarle un valor entre 0 y 1 de acuerdo al impacto que desee tengan los nuevos datos a incorporar.

Una vez introducido el factor de ponderación o de aprendizaje  $w$  y modificadas las ecuaciones recursivas (con la inclusión de dicho factor) se denominó a esa técnica como “*Weighted Adaptive Recursive Principal Component Analysis*” o WARP. A pesar de la existencia de otras técnicas recursivas de detección de fallas basadas en PCA encontradas en la literatura, la complejidad computacional de WARP resultó ser menor que la misma para dos técnicas de la literatura con las que se comparó, tal como se evidencia en la *Tabla 1* y en la *Figura 10*. El desempeño de la técnica WARP fue evaluado en un proceso que es usado comúnmente para la evaluación de técnicas de detección de fallas, el Tennessee Eastman Process (TEP). Para la evaluación se llevan a cabo actualizaciones recursivas incorporando nuevos datos que



han causado una alta tasa de falsas alarmas usando la técnica WARP, haciéndose un barrido mediante la variación del factor de ponderación o de aprendizaje  $w$  desde 0 hasta 1. A medida que se incrementa el valor de  $w$  se va reduciendo la tasa de falsas alarmas luego de la actualización recursiva, tal como se evidencia en la *Figura 35*.

La técnica WARP fue diseñada para abordar cambios en la estructura estadística subyacente a los procesos, sin embargo, las falsas alarmas causadas por variaciones leves del proceso (aquellas que no cambian su estructura estadística) y por la incertidumbre natural de los métodos estadísticos, como es el caso de PCA, deben ser evitadas también. Se diseñó e implementó entonces una técnica, también basada en PCA, llamada “*Moving Average Filter-PCA*” o MAF-PCA, la cual consta de una ventana móvil o media móvil implementada directamente al estadístico  $T^2$ , así como también para el estadístico  $Q$ . Además de la ventana móvil, se diseña, para ambos casos, un algoritmo de filtro que decida cuál será el umbral para cada instante de muestro basándose en una observación de la media y desviación estándar de la ventana. El algoritmo de MAF-PCA también fue evaluado en el Tennessee Eastman Process (TEP), reduciendo con éxito la tasa de falsas alarmas, tal como se evidencia en la *Figura 38* y la *Figura 39*.

Las técnicas diseñadas e implementadas en esta investigación mostraron un desempeño satisfactorio al reducir la tasa de falsas alarmas en un proceso de validación (TEP), cumpliendo de esta manera con los objetivos planteados inicialmente.

## REFERENCIAS

- [1] R. Isermann and P. Ballé. Trends in the application of model based fault detection and diagnosis of technical processes. *In Proc. of the 13th IFAC World Congress, volume N, pages 1-12, Piscataway, NJ, 1996. IEEE Press.*
- [2] R. Isermann. Model based fault detection and diagnosing methods. *In Proc. of the American Control Conf., pages 1605-1609, Piscataway, NJ, 1995. IEE Press.*
- [3] E. L. Russell, L. H. Chiang and R. D. Braatz. *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. Great Britain: Springer-Verlag, 2000.
- [4] P. Burrell and D. Inman. An expert system for the analysis of faults in an electricity supply network: Problems and achievements. *Computers in industry*, 37(2):113-123, 1997.
- [5] J.C. Jeng. Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms. *Journal of the Taiwan Institute of Chemical Engineers* 41 (2010) 475–481. 2010.
- [6] A.K. Samantaray, K. Medjaher, B. Ould Bouamama, M. Staroswiecki, G. Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. *Department of Mechanical Engineering, Indian Institute of Technology, 721 302 Kharagpur, India.* 2005.
- [7] Torres, Melitsa J. Diseño de un método híbrido de detección de fallas y reconocimiento de patrones para procesos industriales, M.S. Tesis, Universidad del Norte, Barranquilla, Colombia, 2012.
- [8] J.F. McGregor and T. Kourti. Statistical process control of multivariate processes. *Control Eng. Practice*, Vol. 3, No. 3, pp. 403-414, 1995.
- [9] K. Amanian, K Salahshoor, M.R. Jafari, M. Mosallaei. Soft Sensor Based on Dynamic Principal Component Analysis and Radial Basis Function Neural Network for Distillation Column. *Proceedings of the World Congress on Engineering and Computer Science 2007 WCECS 2007, October 24-26, 2007, San Francisco, USA.*
- [10] S.W. Choi, I.B. Lee, Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chemical Engineering Science* 59 (2004) 5897 – 5908.
- [11] L.J. Chiang, R.D. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems* 65 (2003) 159– 178.
- [12] Q. Chen, U. Kruger, M. Meronk, A.Y.T. Leung. Synthesis of  $T^2$  and  $Q$  statistics for process monitoring. *Control Engineering Practice* 12 (2004) 745–755.

- [13]G. Stefatos, A.B. Hamza. Dynamic independent component analysis approach for fault detection and diagnosis. *Expert Systems with Applications* 37 (2010) 8606–8617.
- [14]Yong Xu, Jing-Yu Jang, Jiang Yang. A reformative kernel Fisher discriminant analysis. *Pattern Recognition* 37 (2004) 1299 – 1302.
- [15]Posada J.D., Torres M.J., Sanjuán M.E. [2011]. “Fault Detection Implementation for a Regional Natural Gas Transportation System”. Proceedings of the 2011 IASTED International Conference on Power and Energy Systems and Applications, Pittsburgh, USA, November 7-9, 2011, ISBN: 978-0-88986-905-9.
- [16]Trendafilova I. An automated procedure for detection and identification of ball bearing damage using multivariate statistics and pattern recognition. *Mechanical Systems and Signal Processing* 24 (2010) 1858–1869.
- [17]Zarei J. Induction motors bearing fault detection using pattern recognition techniques. *Expert Systems with Applications* 39 (2012) 68–73.
- [18]Martins J.F., Pires V.F., Amaral T. Induction motor fault detection and diagnosis using a current state space pattern recognition. *Pattern Recognition Letters* 32 (2011) 321–328.
- [19]Chiang L., Russell E., Braatz R. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 50 (2000) 243–252.
- [20]Weihua L., H. Yue, S. Valle-Cervantes, S.J. Qin. Recursive PCA for adaptive process monitoring. *Journal of Process Control* 10 (2000) 471-486.
- [21]P.M. Frank. Robust model-based fault detection in dynamic systems. In P.S. Dhurjati and G. Stephanopoulos, editors, *On-line Fault Detection and Supervision in the Chemical Process Industries*, pages 1-13. Pergamon Press, Oxford, 1993. IFAC Symposia Series, Number 1.
- [22]R. K. Mehra, J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637-640, 1971.
- [23]J.F. McGregor and T. Kourti. Statistical process control of multivariate processes. *Control Eng. Practice*, Vol. 3, No. 3, pp. 403-414, 1995.
- [24]K. Amanian, K Salahshoor, M.R. Jafari, M. Mosallaei. Soft Sensor Based on Dynamic Principal Component Analysis and Radial Basis Function Neural Network for Distillation Column. *Proceedings of the World Congress on Engineering and Computer Science 2007 WCECS 2007, October 24-26, 2007, San Francisco, USA*.
- [25]V.B. Gallagher, R.M. Wise, S.W. Butler, D.D. White, G.G. Barna. Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process; improving robustness through model updating, in: *Proc. Of ADCHEM 97. Banff, Canada, 9-11 June 1997, pp. 78-83*.

- [26]T. Iwashita. Asymptotic null and nonnull distribution of Hotelling's  $T^2$ -statistic under the elliptical distribution. *Journal of Statistical Planning and Inference* 61 (1997) 85-104.
- [27]Y-S.P. Chiu, H-D. Ling. A hybrid approach based on Hotelling statistics for automated visual inspection of display blemishes in LCD panels. *Expert Systems with Applications* 36 (2009) 12332–12339.
- [28]K.C. Chow, K-J. Tan, H. Tabe, J. Zhang, N.F Thornhill. Dynamic Principal Component Analysis Using Integral Transforms. *AIChE Annual Meeting, Miami Beach, November 1999, Paper No, 232(c)*.
- [29]S. Wold. Exponentially weighted moving principal component analysis and projection to latent structures, *Chemometrics and Intelligent Laboratory systems* 23 (1994) 149-161.
- [30]Yingwei Zhang., Shuai Li., Yongdong Teng. Dynamic processes monitoring using kernel principal component analysis. *Chemical Engineering Science* 72 (2012) 78-86.
- [31]Weihua Li. J. Qin. Consistent dynamic PCA based on errors-in-variables subspace identification. *Journal of Process Control* 11 (2001) 661-668.
- [32]K. Wenfu, R. Storer, C. Gerogakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligence Laboratory Systems* 30 (1995) 179-196.
- [33]P. Maravelakis, P. Castagliola. EWMA chart for monitoring the process standard deviation when parameters are estimated. *Computational Statistics and Data Analysis* 53 (2009) 2653-2664.
- [34]M.A. Bin Shams, H.M. Budman, T.A. Duever. Fault detection, identification and diagnosis using CUSUM based PCA. *Chemical Engineering Science* 66 (2011) 4488-4498.
- [35]D.C. Montgomery. *Introduction to Statistical Quality Control*. John Wiley and Sons, New York, 1985.
- [36]J.R. Bunch, C.P. Nielsen, D.C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 1978/79, Volume 31, Issue 1, pp 31-48.
- [37]Downs J.J., Vogel E.F. (1993). A plant-wide industrial process control problem. *Comp. Chem. Eng.*, 17(3), 245-255.
- [38]S. F. Yang, J. S. Lin, S. W. Cheng. A new nonparametric EWMA Sign Control Chart. *Expert Systems with Applications* 38 (2011) 6239-6243.
- [39]A. Alkaya, I. Eker. Variance sensitive adaptive threshold-based PCA method for fault detection and diagnosis. *ISA Transactions* 50 (2011) 287-302.
- [40]S.J. Qin. Statistical process monitoring: Basics and beyond. *J Chemom* 2003; 17 (8-9): 480-502.

- [41]X. Liu, U. Kruger, L. Xie, S. Wang. Moving window kernel PCA for adaptive monitoring of nonlinear process. *Chemom Intell Lab Syst* 2009; 96 (2): 132-43.
- [42]S.J. Qin. Recursive PLS algorithms for adaptive data modelling. *Comput Chem Eng* 1998; 22 (4-4): 503-14.
- [43]M. Elshenawy, Y. Shen, S. Naik, X. Ding. Efficient recursive PCA algorithms for process monitoring. *Ind Eng Chem Res.* 49 (2010) 252-259.
- [44]K. Salahshoor, M. Kordestani, M.S. Khoshro. Fault detection and diagnosis of an industrial steam turbine using fusion of SVM (support vector machine) and ANFIS (adaptive neuro-fuzzy inference system) classifier. *Energy* 35 (2010) 5472-5482.
- [45]V. Patel, A. Patel. Recent Advances In AI Based Intelligent Fault Technique For Rolling Element Bearing – A Review. *International Journal of Artificial Intelligence and Mechatronics*, Volume 2, Issue 4, (104-108) ISSN 2320-5121.
- [46]L.J. de Miguel, L.F. Blázquez. Fuzzy logic-based decision-making for fault diagnosis in a DC motor. *Engineering Applications* 18 (2005) 423-450.
- [47]A. Lemos, W. Caminhas. F. Gomide. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information Sciences* 220 (2013) 64-85.
- [48]Z.N. Sadough Vanini, K. Khorasani, N. Meskin. Fault detection and isolation of a dual spool gas turbine using the dynamic neural networks and multiple model approach. *Information Sciences* 259 (2014) 234-251.
- [49]V. Venkatasubramanian, R. Rengasamy, S.N. Kavuri, K. Yin. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computer & Chemical Engineering.* 27 (2003) 327-346.
- [50]B. Samanta, K.R. Al-Balushi, S.A. Al-Araimi. Artificial neural network and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence* 16 (2003) 657-665.
- [51]S.R. Samantaray, P.K. Dash, S.K. Upadhyay. Adaptive Kalman filter and neural network based high impedance fault detection. *Electrical Power and Energy Systems* 31 (2009) 167-172.
- [52]M. Staroswiecki, G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica* 37 (2001) 687-699.
- [53]P.M. Frank. Analytical and Qualitative Model-based Fault Diagnosis – A Survey and Some New Results. *European Journal of Control* (1996) 2:6-28.
- [54]J. Montmain, S. Gentil. Dynamic causal model diagnostic reasoning for online technical process supervision. *Automatica* 36 (2000) 1137-1152.

## ANEXOS

### A.1. Esquema del Proceso

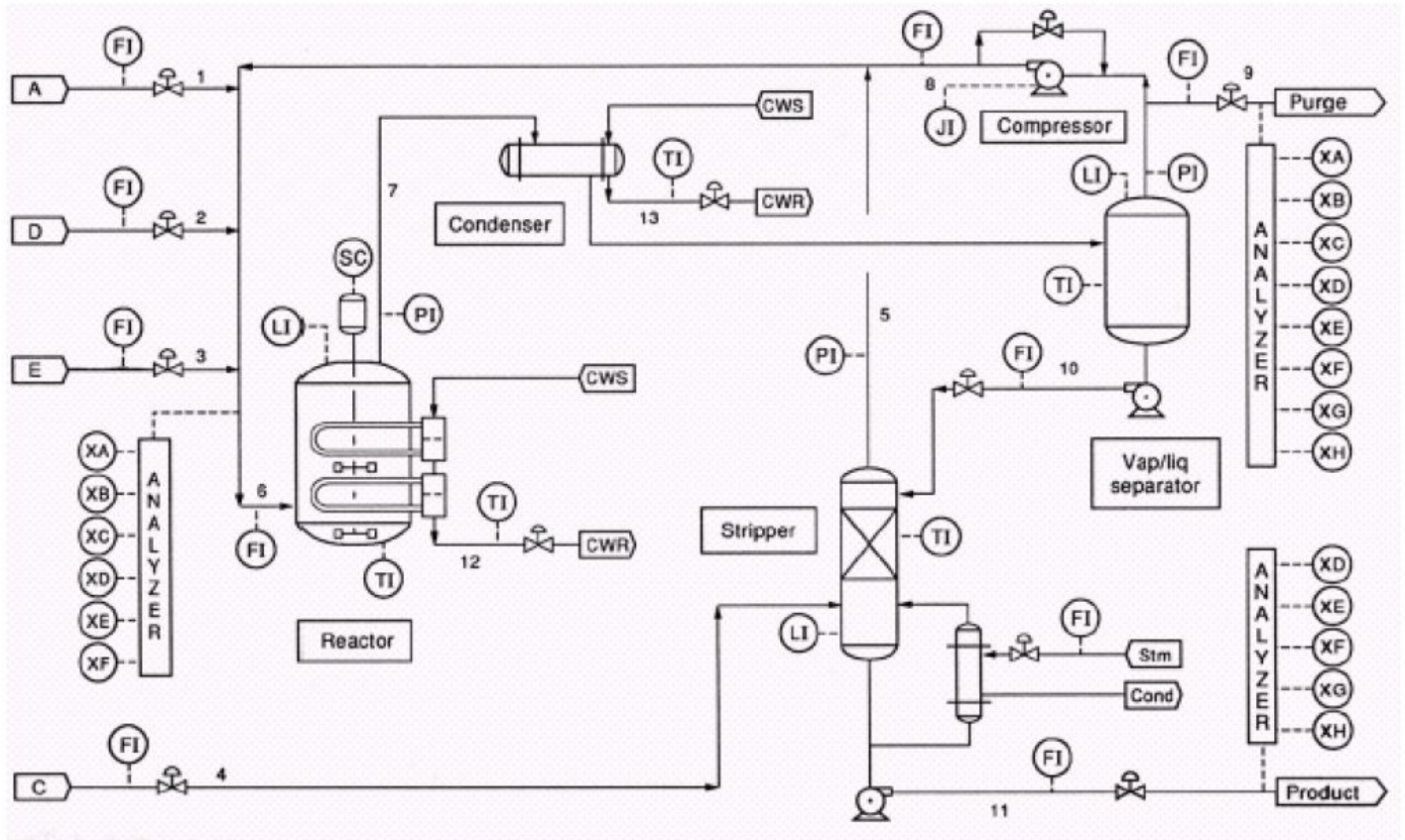


Figura A 1 –Esquema del proceso (Tennessee Eastman Process).

### A.2. Modelo del Proceso

Las ecuaciones que modelan el Tennessee Eastman Process se pueden dividir en ecuaciones de estado y en ecuaciones de salida.

### A.2.1. Ecuaciones de Estado

Se muestra a continuación las ecuaciones de estado del TEP.

$$\frac{dN_{i,m}}{dt} = \sum_{j=1,2,3,5,8} y_{i,j} F_j - y_{i,6} F_6 \quad (i = A, B, \dots, H) \quad (A.1)$$

$$\frac{dN_{i,r}}{dt} = y_{i,6} F_6 - y_{i,7} F_7 + \sum_{j=1}^3 V_{i,j} R_j \quad (i = A, B, \dots, H) \quad (A.2)$$

$$\frac{dN_{i,s}}{dt} = y_{i,7} F_7 - y_{i,8} (F_8 + F_9) - x_{i,10} F_{10} \quad (i = A, B, \dots, H) \quad (A.3)$$

$$\frac{dN_{i,p}}{dt} = (1 - \phi_i) (x_{i,10} F_{10} + y_{i,4} F_4) - x_{i,11} F_{11} \quad (i = G, H) \quad (A.4)$$

$$\left( \sum_{i=A}^H N_{i,m} C_{p,vap,i} \right) \frac{dT_m}{dt} = \sum_{j=1,2,3,5,8} F_j \left( \sum_{l=A}^H y_{l,j} C_{p,vap,l} \right) (T_j - T_m) \quad (A.5)$$

$$\left( \sum_{i=A}^H N_{i,r} C_{p,i} \right) \frac{dT_r}{dt} = F_6 \left( \sum_{l=A}^H y_{l,6} C_{p,vap,l} \right) (T_6 - T_r) - \dot{Q}_r - \sum_{j=1}^3 \Delta H_{Rj} R_j \quad (A.6)$$

$$\left( \sum_{i=A}^H N_{i,s} C_{p,i} \right) \frac{dT_s}{dt} = F_7 \left( \sum_{l=A}^H y_{l,7} C_{p,vap,l} \right) (T_r - T_s) - H_0 V_s - \dot{Q}_s \quad (A.7)$$

$$\begin{aligned} & \left( \sum_{i=A}^H N_{i,p} C_{p,i} \right) \frac{dT_p}{dt} \\ &= F_{10} \left( \sum_{l=A}^H x_{l,10} C_{p,l} \right) (T_s - T_p) + F_4 \left( \sum_{l=A}^H y_{l,4} C_{p,vap,l} \right) (T_4 - T_p) \\ & - H_0 V_p + \dot{Q}_p \quad (A.8) \end{aligned}$$

### A.2.2. Ecuaciones de las Variables de Salida

Se muestran a continuación las ecuaciones de salida del TEP.

$$P_m = \sum_{i=A}^H N_{i,m} \frac{RT_m}{V_m} \quad (A.9)$$

$$y_{i,6} = \frac{N_{i,m}}{\sum_{i=A}^H N_{i,m}} \quad (i = A, B, \dots, H) \quad (A.10)$$

$$P_r = \sum_{i=A}^H P_{i,r} \quad (A.11)$$

donde

$$P_{i,r} = \frac{N_{i,r}RT_r}{V_{V_r}} \quad (i = A, B, C) \quad (A.12)$$

$$P_{i,r} = \gamma_{i,r} x_{i,r} P_i^{sat}(T_r) \quad (i = D, E, \dots, H) \quad (A.13)$$

$$P_i^{sat} = 1 \times 10^{-3} \exp \left( A_i + \frac{B_i}{C_i + T_r - T^*} \right) \quad (i = D, E, \dots, H) \quad (A.14)$$

$$F_6 = 0.8334 \frac{kmol}{s\sqrt{MPa}} \sqrt{\|P_m - P_r\|} \quad (A.15)$$

$$F_7 = 1.5344 \frac{kmol}{s\sqrt{MPa}} \sqrt{\|P_r - P_s\|} \quad (A.16)$$

$$\Delta H_{Rj} = \sum_{i=A}^H H_i V_{i,j} + H_0 F_j, \quad \text{with } H_i = C_{p,i}(T_r - T^*) \quad (j = 1, 2, 3) \quad (A.17)$$

$$\dot{Q}_r = m_{CW,r} C_{p,CW} (T_{CW,r,out} - T_{CW,r,in}) \quad (A.18)$$

$$\dot{Q}_r = UA_r \left[ \frac{\Delta T_{1,r} - \Delta T_{2,r}}{\ln(\Delta T_{1,r}/\Delta T_{2,r})} \right] \quad (A.19)$$



donde

$$\Delta T_{1,r} = T_r - T_{CW,r,in}, \quad \Delta T_{2,r} = T_r - T_{CW,r,out} \quad (A.20)$$

$$x_{i,r} = 0 \quad (i = A, B, C) \quad (A.21)$$

$$x_{i,r} = \frac{N_{i,r}}{\sum_{i=D}^H N_{i,r}} \quad (i = D, E, \dots, H) \quad (A.22)$$

y

$$y_{i,7} = \frac{P_{i,r}}{P_r} \quad (i = A, B, \dots, H) \quad (A.23)$$

$$V_{1r} = \sum_{i=D}^H \frac{N_{i,r}}{\rho_i} \quad (A.24)$$

$$V_{V_r} = V_r - V_{1r} \quad (A.25)$$

$$P_s = \sum_{i=A}^H P_{i,s} \quad (A.26)$$

donde

$$P_{i,s} = \frac{N_{i,s} R T_s}{V_{V_s}} \quad (i = A, B, C) \quad (A.27)$$

$$P_{i,s} = \gamma_{is} x_{i,10} P_i^{sat}(T_s) \quad (i = D, E, \dots, H) \quad (A.28)$$

$$T_8 = T_s \left( \frac{P_m}{P_s} \right)^{\frac{1-k}{k}} \quad (A.29)$$

$$H_0 V_s = \sum_{i=D}^H x_{i,10} F_{10} \cdot H_{vap,i} \quad (A.30)$$

$$\dot{Q}_s = m_{CW,s} C_{p,CW} (T_{CW,s,out} - T_{CW,s,in}) \quad (A.31)$$

$$\dot{Q}_s = U A_s \left[ \frac{\Delta T_{1,s} - \Delta T_{2,s}}{\ln(\Delta T_{1,s} / \Delta T_{2,s})} \right] \quad (A.32)$$

donde

$$\Delta T_{1,s} = T_s - T_{CW,s,in}, \quad \Delta T_{2,s} = T_s - T_{CW,s,out} \quad (A.33)$$

$$y_{i,8} = y_{i,9} = \frac{P_{i,s}}{P_s} \quad (i = A, B, \dots, H) \quad (A.34)$$

$$x_{i,10} = 0 \quad (i = A, B, C) \quad (A.35)$$

$$x_{i,10} = \frac{N_{i,s}}{\sum_{i=D}^H N_{i,s}} \quad (i = D, E, \dots, H) \quad (A.36)$$

$$V_{1s} = \sum_{i=D}^H \frac{N_{i,s}}{\rho_i} \quad (A.37)$$

$$V_{V_r} = V_s - V_{1s} \quad (A.38)$$

$$H_0 V_p = \sum_{i=D}^H (y_{i,5} F_5 - y_{i,4} F_4) \cdot H_{vap,i} \quad (A.39)$$

$$\dot{Q}_p = 2258.717 \frac{kJ}{kg} \dot{m}_{steam} \quad (A.40)$$

$$V_{L_p} = \sum_{i=D}^H \frac{N_{i,p}}{\rho_i} \quad (A.41)$$

$$\phi_i = \sum_{j=0}^3 a_{i,j} (T_s - 273) \quad (i = D, E, \dots, H) \quad (A.42)$$

$$\phi_i = 1 \quad (i = A, B, C) \quad (A.43)$$

$$F_s = F_{10} + F_4 - F_{11} - \sum_{i=G}^H \frac{dN_{i,p}}{dt} \quad (A.44)$$

$$y_{i,5} = \frac{\phi_i (y_{i,4} F_4 + x_{i,10} F_{10})}{F_5} \quad (i = A, B, \dots, H) \quad (A.45)$$

$$x_{i,11} = \frac{y_{i,4}F_4 + x_{i,10}F_{10} - y_{i,5}F_5}{F_{11}} \quad (i = D, E, F) \quad (A.46)$$

$$x_{i,11} = \left(1 - \sum_{j=D}^F x_{j,11}\right) \frac{N_{i,p}}{\sum_{j=D}^H N_{j,p}} \quad (i = G, H) \quad (A.47)$$

$$R_1 = \alpha_1 V_{V_r} \exp \left[ 44.06 - \frac{42600}{RT_r} \right] P_{A,r}^{1.08} P_{C,r}^{0.311} P_{D,r}^{0.874} \quad (A.48)$$

$$R_2 = \alpha_2 V_{V_r} \exp \left[ 10.27 - \frac{19500}{RT_r} \right] P_{A,r}^{1.15} P_{C,r}^{0.370} P_{E,r}^{1.00} \quad (A.49)$$

$$R_3 = \alpha_3 V_{V_r} \exp \left[ 59.50 - \frac{59500}{RT_r} \right] P_{A,r} (0.77 P_{D,r} + P_{E,r}) \quad (A.50)$$

### A.3. Datos de Proceso

Se muestran, en la *Tabla A1*, las propiedades de los componentes del TEP involucrados en los procesos que ocurren en él.

**Tabla A 1 – Propiedades de los componentes del TEP.**

Componente	Peso Molecular	Densidad de Líquido ( $kg/m^3$ )	Capacidad Calorífica de Líquido ( $kJ/kg^\circ C$ )	Capacidad Calorífica de Vapor ( $kJ/kg^\circ C$ )	Calor de Vaporización ( $kJ/kg$ )	Presión de Vapor (Ecuación de Antoine)		
						A	B	C
A	2	-	-	14,6	-	-	-	-
B	25,4	-	-	2,04	-	-	-	-
C	28	-	-	1,05	-	-	-	-
D	32	299	7,66	1,85	202	20,81	-1444	258
E	46	365	4,17	1,87	372	21,24	-2114	266
F	48	328	4,45	2,02	372	21,24	-2144	266
G	62	612	2,55	0,712	523	21,32	-2748	233
H	76	617	2,45	0,628	486	22,1	-3318	250

Se listan en la *Tabla A2* las variables manipuladas o variables de entrada del proceso:

**Tabla A 2 – Variables de entrada del TEP.**

<b>Variable</b>	<b>Descripción</b>
XMV(1)	Flujo de alimentación D (corriente 2)
XMV(2)	Flujo de alimentación E (corriente 3)
XMV(3)	Flujo de alimentación A (corriente 1)
XMV(4)	Flujo de alimentación total (corriente 4)
XMV(5)	Válvula de compresor de reciclo
XMV(6)	Válvula de purgue (corriente 9)
XMV(7)	Flujo de líquido en el separador
XMV(8)	Flujo de producto líquido en el separador (corriente 11)
XMV(9)	Válvula de vapor del separador
XMV(10)	Flujo de agua de enfriamiento en el reactor
XMV(11)	Flujo de agua de enfriamiento en el condensador
XMV(12)	Velocidad del agitador

Se listan en la *Tabla A3* las variables de salida del TEP:

**Tabla A 3 - Variables de salida del TEP.**

<b>Variable</b>	<b>Descripción</b>	<b>Unidades</b>
XMEAS(1)	Alimentación de A (corriente 1)	<i>kscmh</i>
XMEAS(2)	Alimentación de D (corriente 2)	<i>Kg/h</i>
XMEAS(3)	Alimentación de E (corriente 3)	<i>Kg/h</i>
XMEAS(4)	Alimentación total (corriente 4)	<i>kscmh</i>
XMEAS(5)	Flujo de reciclo (corriente 8)	<i>kscmh</i>
XMEAS(6)	Tasa de alimentación del reactor	<i>kscmh</i>
XMEAS(7)	Presión del reactor	<i>kPa</i>
XMEAS(8)	Nivel del reactor	%
XMEAS(9)	Temperatura del reactor	°C
XMEAS(10)	Tasa de purgue (corriente 9)	<i>kscmh</i>
XMEAS(11)	Temperatura de separación de producto	°C
XMEAS(12)	Nivel de separación de producto	%
XMEAS(13)	Presión de separación de producto	<i>kPa</i>

XMEAS(14)	Prod Sep subflujo (Underflow) (corriente 10)	$m^3/h$
XMEAS(15)	Nivel del separador	%
XMEAS(16)	Presión del separador	$kPa$
XMEAS(17)	Subflujo (Underflow) del Separador (corriente 11)	$m^3/h$
XMEAS(18)	Temperatura del separador	$^{\circ}C$
XMEAS(19)	Flujo de vapor en el separador	$Kg/h$
XMEAS(20)	Trabajo del compresor	$kW$
XMEAS(21)	Temperatura de salida de agua de enfriamiento del reactor	$^{\circ}C$
XMEAS(22)	Temperatura de salida de agua de enfriamiento del separador	$^{\circ}C$
XMEAS(23)	Concentración de A en la corriente 6	$mol\%$
XMEAS(24)	Concentración de B en la corriente 6	$mol\%$
XMEAS(25)	Concentración de C en la corriente 6	$mol\%$
XMEAS(26)	Concentración de D en la corriente 6	$mol\%$
XMEAS(27)	Concentración de E en la corriente 6	$mol\%$
XMEAS(28)	Concentración de F en la corriente 6	$mol\%$
XMEAS(29)	Concentración de A en la corriente 9	$mol\%$
XMEAS(30)	Concentración de B en la corriente 9	$mol\%$
XMEAS(31)	Concentración de C en la corriente 9	$mol\%$
XMEAS(32)	Concentración de D en la corriente 9	$mol\%$
XMEAS(33)	Concentración de E en la corriente 9	$mol\%$
XMEAS(34)	Concentración de F en la corriente 9	$mol\%$
XMEAS(35)	Concentración de G en la corriente 9	$mol\%$
XMEAS(36)	Concentración de H en la corriente 9	$mol\%$
XMEAS(37)	Concentración de D en la corriente 11	$mol\%$
XMEAS(38)	Concentración de E en la corriente 11	$mol\%$
XMEAS(39)	Concentración de F en la corriente 11	$mol\%$
XMEAS(40)	Concentración de G en la corriente 11	$mol\%$
XMEAS(41)	Concentración de H en la corriente 11	$mol\%$

#### A.4. Nomenclatura

Se muestra a continuación la notación usada en las ecuaciones previamente definidas para el modelamiento del Tennessee Eastman Process.

$C_{p,cw}$ : Capacidad calorífica específica del agua,  $kJ\ kg^{-1}K^{-1}$

$C_{p,i}$ : Capacidad calorífica específica del componente  $i$  en fase líquida,  $kJ\ kg^{-1}K^{-1}$

$C_{p,vap,i}$ : Capacidad calorífica específica del componente  $i$  en fase gaseosa,  $kJ\ kg^{-1}K^{-1}$

$F_j$ : Flujo molar de la corriente  $j$ ,  $kmol\ h^{-1}$

$H_i$ : Entalpía del componente  $i$ ,  $kJ$

$H_0$ : Entalpía de referencia,  $kJ$

$\Delta H_{Rj}$ : Calor exotérmico,  $kJ\ kmol^{-1}$

$m_{cw,r}$ : Flujo de agua de enfriamiento del reactor,  $kg\ h^{-1}$

$m_{cw,s}$ : Flujo de agua de enfriamiento del separador,  $kg\ h^{-1}$

$N_{ik}$ : Contenido molar total del componente  $i$  en la unidad  $k$  ( $k = m, r, s, p$ ),  $kmol$

$P_{i,j}$ : Presión parcial del componente  $i$  en la unidad  $k$  ( $k = m, r, s, p$ ),  $kPa$

$P_i^{sat}$ : Presión de saturación del componente  $i$ ,  $kPa$

$P_k$ : Presión total en la unidad  $k$  ( $k = m, r, s, p$ ),  $kPa$

$Q_k$ : Energía removida de la unidad  $k$  ( $k = r, s, p$ ),  $kW$

$R_i$ : Conversión del componente  $i$ ,  $kmol\ h^{-1}$

$R$ : Constante de gas

$t$ : Tiempo,  $h$

$T_{cw,k,in}$ : Temperatura de entrada de agua de enfriamiento en la unidad  $k$  ( $k = r, s$ ),  $K$

$T_{cw,k,out}$ : Temperatura de salida de agua de enfriamiento en la unidad  $k$  ( $k = r, s$ ),  $K$

$T_k$ : Temperatura de la unidad  $k$  ( $k = m, r, s, p$ ),  $K$

$T^*$ : Temperatura absoluta,  $K$

$UA$ : Tasa de transferencia de calor específico,  $kW\ K^{-1}$

$V_k$ : Volumen total de la unidad  $k$  ( $k = m, r, s, p$ ),  $m^3$

$V_{L,k}$ : Volumen de líquido en la unidad  $k$  ( $k = m, r, s, p$ ),  $m^3$

$V_{V,k}$ : Volumen de vapor en la unidad  $k$  ( $k = m, r, s, p$ ),  $m^3$

$x_{i,k}$ : Fracción molar en fase líquida del componente  $i$  en la unidad  $k$  ( $k = m, r, s, p$ )

$y_{i,k}$ : Fracción molar en vapor del componente  $i$  en la unidad  $k$  ( $k = m, r, s, p$ )

#### *Subíndices*

$r$ : Reactor

$s$ : Separador

$m$ : Zona de mezclado

$p$ : Producto (en el separador)

$L$ : Líquido

$V$ : Vapor

$i$ : Componente

$j$ : Corriente

#### *Símbolos griegos*

$\alpha$ : Parámetro ajustable usado en ecuaciones de tasa de reacción.

$\gamma$ : Coeficiente de actividad

$\phi$ : Factor de separación

$v$ : Coeficiente estequiométrico.

$\rho$ : Densidad molar,  $mol\ m^{-3}$

#### A.4. Código del Algoritmo WARP en MatLab®

```
clear all
clc
b0=mean(X0); variance0=var(X0);
[n0,m0]=size(X0);
[n_1,m_1]=size(X_1);
b_1=mean(X_1); variance_1=var(X_1);
w=0.5;
X1=[X0' X_1']'; [n1,m1]=size(X1); b1=w*b0+(1-w)*b_1;
Xob=[X1' X_1' X_1']';
I_1=ones(n_1,1); b1_f=mean(X1); delta_b=b1-b0;
for i=1:m1
    varianza0(i,i)=variance0(i);
    varianza_1(i,i)=variance_1(i);
end
varianza1=w*varianza0+(1-w)*varianza_1+(w+(1/(n1-1)))*(delta_b')*delta_b;
for i=1:m1
    variancel(i)=varianza1(i,i);
end
for k=1:n0
    for i=1:m0
        X0(k,i)=(X0(k,i)-b0(i))/((variance0(i))^0.5);
        Sigma0(i,i)=(variance0(i))^0.5;
    end
end
S0=(1/(n0-1))*(X0')*X0;
[V0,D0]=eig(S0); Sig0=D0^0.5;
for k=1:n_1
    for i=1:m_1
        X_1(k,i)=(X_1(k,i)-b1(i))/((variancel(i))^0.5);
    end
end
S_1=(1/(n_1-1))*(X_1')*X_1;
for k=1:n1
    for i=1:m1
        X1(k,i)=(X1(k,i)-b1(i))/((variancel(i))^0.5);
        Sigma1(i,i)=(variancel(i))^0.5;
    end
end
S1_pred=w*(Sigma1^-1)*Sigma0*S0*Sigma0*(Sigma1^-1)+(1-w)*S_1+(w+(1/(n1-1)))*(Sigma1^-1)*(delta_b')*delta_b*(Sigma1^-1);
D_tilda1=(w+(1/(n1-1)))*D0; alfa1=1; Omega1=[((1-w)/(n_1-1))^0.5*X_1' ((w+(1/(n1-1)))^0.5)*(Sigma1^-1)*delta_b'];
z1=(V0')*Omega1;
[R1,DD1]=eig(D_tilda1+alfa1*z1*z1');
D1_pred=DD1; Sig1_pred=D1_pred^0.5;
V1_pred=V0*R1;
```



```

for k=1:n1
    for i=1:m1
        Xob(k,i)=(Xob(k,i)-b0(i))/((variance0(i))^0.5);
    end
end
for k=n1+1:n1+2*n_1
    for i=1:m0
        Xob(k,i)=(Xob(k,i)-b1(i))/((variance1(i))^0.5);
    end
end

for i=1:m0
    d(i)=D0(i,i);
end
d0=d;
D=D0;
V=V0;
for i=1:m0
    for j=1:m0
        if d0(i)<d0(j)
            d(i)=d0(j);
            d(j)=d0(i);
            d0=d;
            D(i,i)=D0(j,j);
            D(j,j)=D0(i,i);
            D0=D;
            V(:,i)=V0(:,j);
            V(:,j)=V0(:,i);
            V0=V;
        end
    end
end
for i=1:m0
    D0(i,i)=D(m0-i+1,m0-i+1);
    V0(:,i)=V(:,m0-i+1);
end
SumD0=0;
for i=1:m0
    SumD0=SumD0+D0(i,i);
end
cc=0; a0=0;
for i=1:m0
    cc=cc+D0(i,i);
    if (cc/SumD0)<0.95
        a0=a0+1;
    end
end
for j=1:a0
    P0(:,j)=V0(:,j);
    Sig00(j,j)=Sig0(j,j);
end
for i=1:m1

```

```

        d(i)=Dl_pred(i,i);
    end
    dl_pred=d;
    D=Dl_pred;
    V=Vl_pred;
    for i=1:m1
        for j=1:m1
            if dl_pred(i)<dl_pred(j)
                d(i)=dl_pred(j);
                d(j)=dl_pred(i);
                dl_pred=d;
                D(i,i)=Dl_pred(j,j);
                D(j,j)=Dl_pred(i,i);
                Dl_pred=D;
                V(:,i)=Vl_pred(:,j);
                V(:,j)=Vl_pred(:,i);
                Vl_pred=V;
            end
        end
    end
    for i=1:m1
        Dl_pred(i,i)=D(m0-i+1,m0-i+1);
        Vl_pred(:,i)=V(:,m0-i+1);
    end
    SumDl=0;
    for i=1:m1
        SumDl=SumDl+Dl_pred(i,i);
    end
    cc=0; a1=0;
    for i=1:m1
        cc=cc+Dl_pred(i,i);
        if (cc/SumDl)<0.95
            a1=a1+1;
        end
    end
    for j=1:a1
        Pl_pred(:,j)=Vl_pred(:,j);
        Sigl_pred1(j,j)=Sigl_pred(j,j);
    end
    Sigl_pred1=real(Sigl_pred1);
    for i=1:n1+1
        Tsquare(i)=Xob(i,:)*P0*(Sig00^-2)*(P0')*Xob(i,:)';
    end
    for i=n1+1:n1+2*n_1
        Tsquare(i)=Xob(i,:)*P1_pred*(Sigl_pred1^-2)*(P1_pred')*Xob(i,:)';
    end
    Tsquare=real(Tsquare);
    Fo=finv(0.95,(a0+1),n0-(a0+1));
    for i=1:n1+1
        Tsquare_alfa(i)=(((a0+1)*(n0-1)*(n0+1))/(n0*(n0-(a0+1))))*Fo;
    end
    Fo=finv(0.95,(a1+1),n1-(a1+1));

```

```

for i=n1+1:n1+2*n_1
Tsquare_alfa(i)=(((a1+1)*(n1-1)*(n1+1))/(n1*(n1-(a1+1))))*Fo;
end
toc
figure(1)
plot(Tsquare_alfa,'r')
hold on
plot(Tsquare)
counter=0; div=0;
for i=n1+1:n1+2*n_1
    div=div+1;
    if Tsquare(i)>Tsquare_alfa(i)
        counter=counter+1; r(i)=1;
    end
end
false_alarm_rate=counter*100/(div);

```

## A.5. Código del algoritmo MAF-PCA en MatLab®

```

clear all
clc
XMEAS=xlsread('X_2.xlsx'); b=mean(XMEAS); variance=var(XMEAS);
[n0,m0]=size(XMEAS);
for i=1:m0
    STD(i)=variance(i)^0.5;
end
XMEAS_1=xlsread('X_3.xlsx'); b_1=mean(XMEAS_1);
variance_1=var(XMEAS_1);
[n_1,m_1]=size(XMEAS_1);
for i=1:m_1
    STD_1(i)=variance_1(i)^0.5;
end
%Xt Training set
[rows,cols]=size(Xt);
n=rows;
m=cols;
B=sum(Xt);
v_mean=B/n; %Sample mean Vector*c
for k=1:m %Training set pretreatment
    for i=1:n
        SQR(i)=(Xt(i,k)-v_mean(k))^2;
    end
    C(k)=sum(SQR);
end
VAR=(C/(n-1));
for i=1:m
    STD(i)=(VAR(i))^0.5;
end

```

```

        end
        for k=1:n
            for i=1:m
                X(k,i)=(Xt(k,i)-v_mean(i))/STD(i); %Pretreated Data
            end
        end
        S=(1/(n-1))*(X')*X; %Sample covariance matrix
        [V,D]=eig(S); %spectral decomposition
        for i=1:m
            d(i)=D(i,i);
        end
        do=d;
        Do=D;
        Vo=V;
        for i=1:m
            for j=1:m
                if d(i)<d(j)
                    do(i)=d(j);
                    do(j)=d(i);
                    d=do;
                    Do(i,i)=D(j,j);
                    Do(j,j)=D(i,i);
                    D=Do;
                    Vo(:,i)=V(:,j);
                    Vo(:,j)=V(:,i);
                    V=Vo;
                end
            end
        end
        for i=1:m
            D(i,i)=Do(m-i+1,m-i+1);
            V(:,i)=Vo(:,m-i+1);
        end
        addo=sum(d);
        add=0;
        i=0;
        while add/addo<=0.95
            i=i+1;
            add=add+D(i,i);
            a=i+1;
        end
        for j=1:a
            P(:,j)=V(:,j);
        end
        T=X*P;
        Xcap=T*P';
        E=X-Xcap;
        Si=D^0.5;
        for i=1:a
            for j=1:a
                Sig(i,j)=Si(i,j);
            end
        end

```

```

end
[rows,cols]=size(Xo);
n=rows;
m=cols;
for k=1:n
    for i=1:m
        Xob(k,i)=(Xo(k,i)-v_mean(i))/STD(i); %Pretreated Data
    end
end
I=zeros(m,m);
k=10;
for i=1:m
    Im(i,i)=1;
end
for i=1:n
    x=Xob(i,:);
    Tsquare(i)=x*P*(Sig^(-2))*(P')*x';
    r=(I-P*P')*x'; Q(i)=(r')*r;
    if i>k
        HT=Tsquare(i-k:i); HQ=Q(i-k:i);
        Tsquare_mean(i)=mean(HT); Q_mean(i)=mean(HQ);
        Tsquare_STD(i)=(var(HT))^0.5; Q_STD(i)=(var(HQ))^0.5;
    end
end
Fo=finv(0.95,a,n-a);
Tsquare_alfa=((a*(n-1)*(n+1))/(n*(n-a)))*Fo;
C_alfa=norminv(0.95,1.67,1.67);
Theta1=0;
for i=a+1:m
    Theta1=Theta1+D(i,i);
end
Theta2=0;
for i=a+1:m
    Theta2=Theta2+D(i,i)^2;
end
Theta3=0;
for i=a+1:m
    Theta3=Theta3+D(i,i)^3;
end
ho=1-(2*Theta1*Theta3)/(3*Theta2^2);
Q_alfa=Theta1*((ho*C_alfa*(2*Theta2)^0.5)/Theta1+1+Theta2*ho*(ho-1)/(Theta1^2))^(1/ho);
prueba=0;
alarms=0; alarms2=0;
for i=1:n
    Tsquare_alfav(i)=Tsquare_alfa; Q_alfav(i)=Q_alfa;
    if i>k
        if (Tsquare_mean(i)+Tsquare_STD(i))>Tsquare_alfav(i)
            if Tsquare(i-2)<=Tsquare_alfa || Tsquare(i-1)<=Tsquare_alfa || Tsquare(i)<=Tsquare_alfa
                Tsquare_alfav(i)=(Tsquare_mean(i)+Tsquare_STD(i))*1.1;
            end
        end
    end
end

```

```

        prueba=prueba+1;
    end
end
end
if Tsquare(i)>Tsquare_alfav(i)
    alarms=alarms+1;
end
if i>k
    if (Q_mean(i)+Q_STD(i))>Q_alfav(i)
        if Q(i-2)<=Q_alfa || Q(i-1)<=Q_alfa || Q(i)<=Q_alfa
            Q_alfav(i)=(Q_mean(i)+Q_STD(i))*1.1;
        end
    end
end
if Q(i)>Q_alfav(i)
    alarms2=alarms2+1;
end
end
plot(Tsquare,'b');
hold on
figure(1)
plot(Tsquare_alfav,'r');
% figure(2)
% plot(Q,'b');
% hold on
% plot(Q_alfav,'r');
Alarms_Rate=(alarms/n)*100;
Alarms_Rate2=(alarms2/n)*100;

```